PRIMA'S OFFICIAL STRATEGY GUIDE

StarCraft™

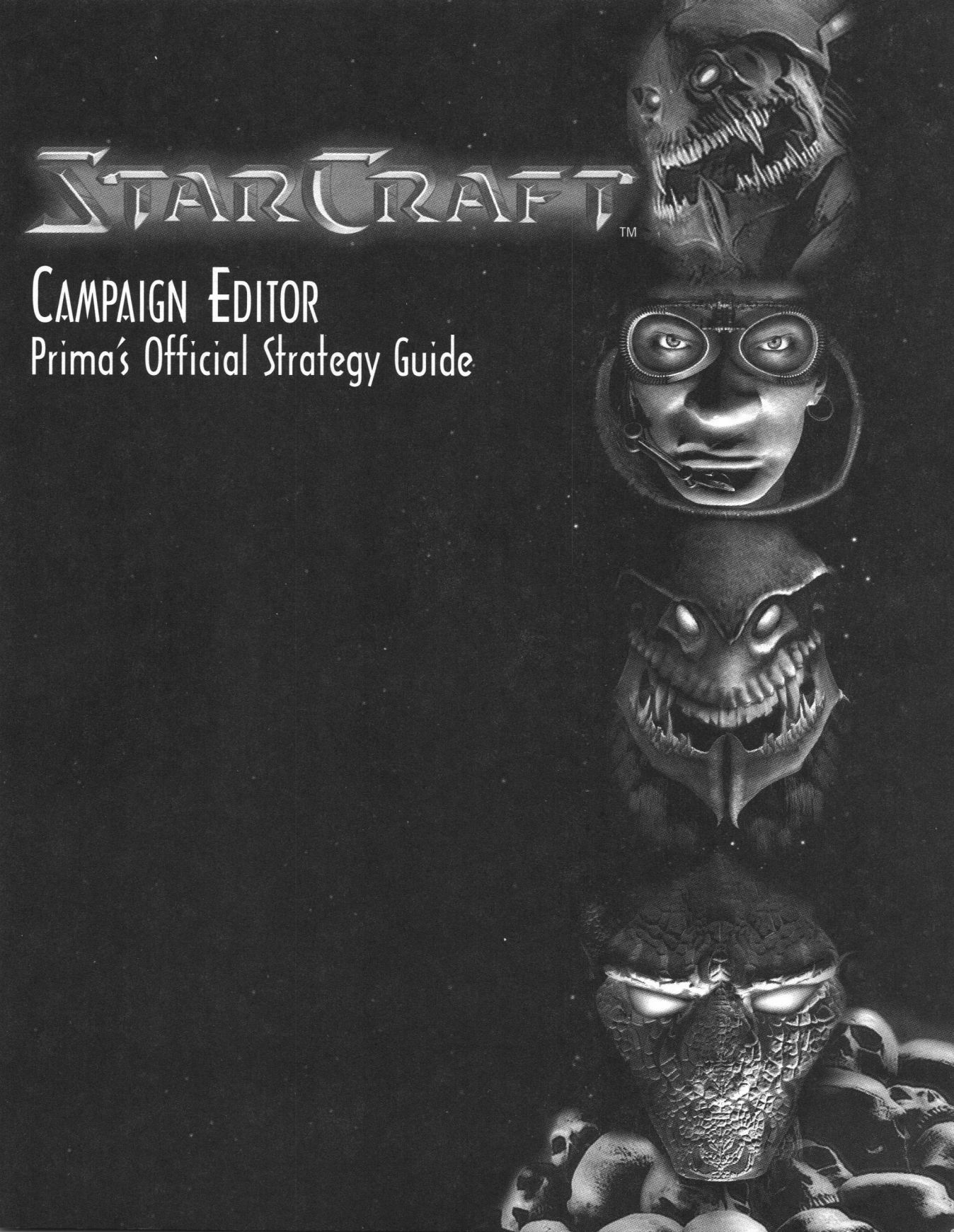
CAMPAIGN EDITOR

CREATE ALL-NEW MISSIONS!

PRIMA®

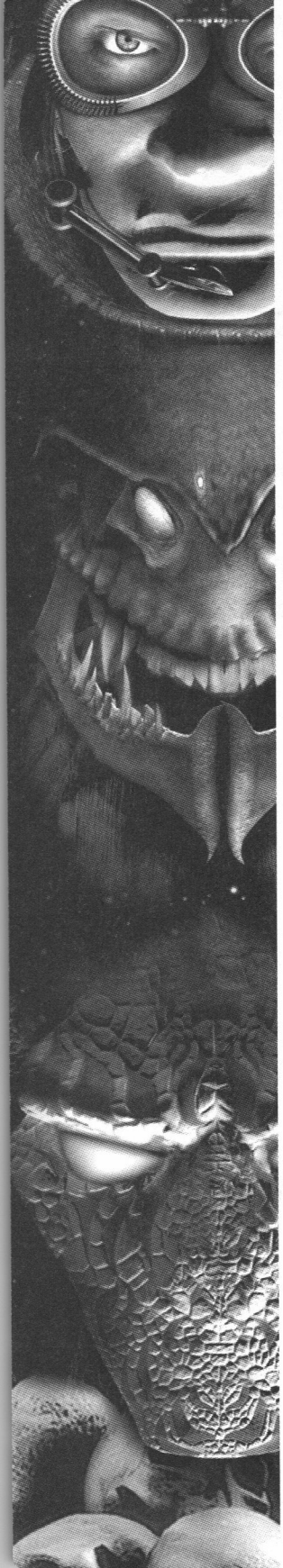
BLIZZARD ENTERTAINMENT

Steve Honeywell
Joe Grant Bell



Campaign Editor

Prima's Official Strategy Guide



® is a registered trademark of Prima Publishing, a division of Prima Communications, Inc.

© 1998 by Prima Publishing. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system without written permission from Prima Publishing, except for the inclusion of quotations in a review.

Project Editor: Sara E. Wilson

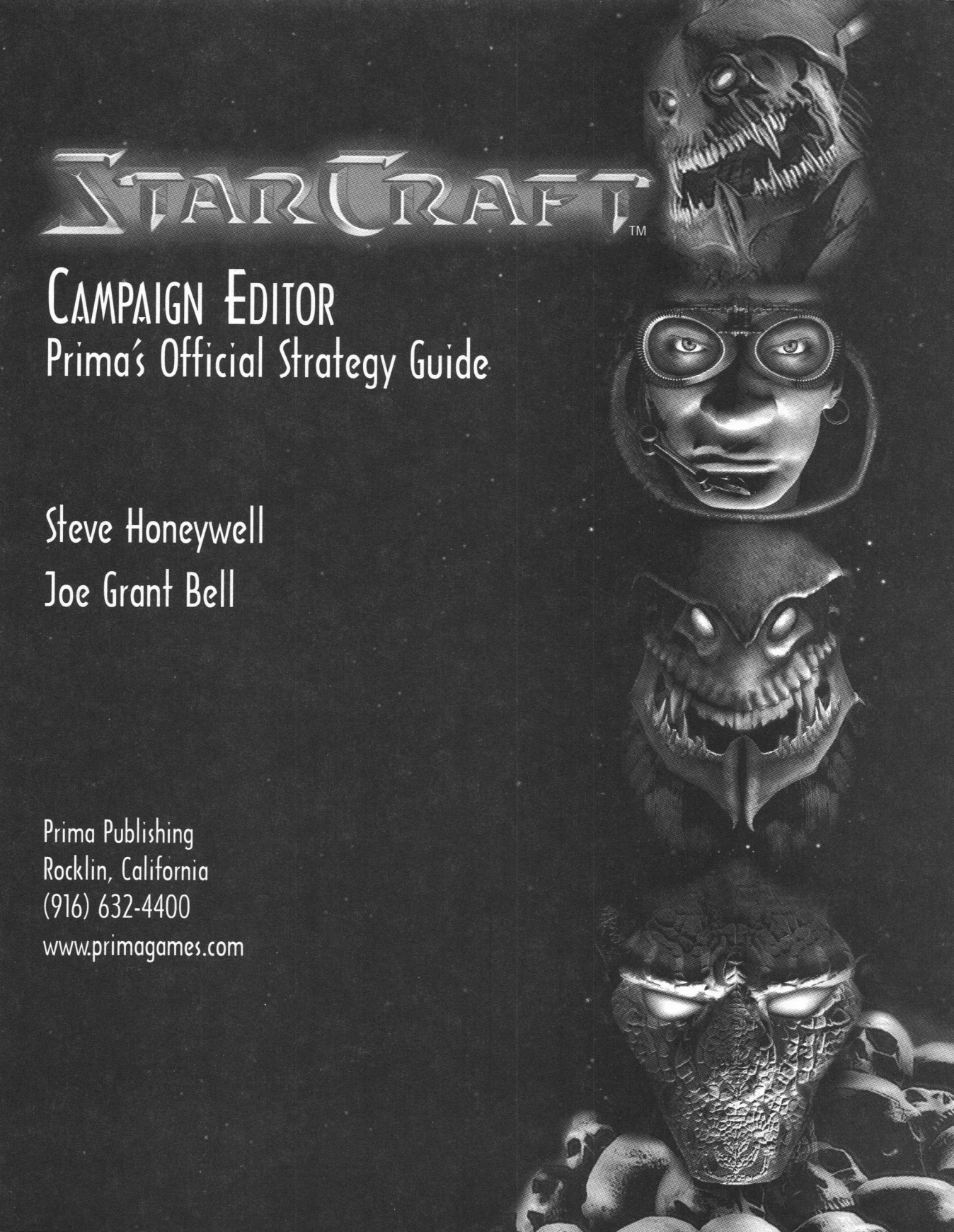
Portions © 1998 Blizzard Entertainment. All rights reserved. Original characters are copyrighted characters of Blizzard Entertainment. Starcraft, Brood War, and Battle.net are trademarks and Blizzard Entertainment is a trademark or registered trademark of Davidson & Associates, Inc. in the U.S. and/or other countries.

All products and characters mentioned in this book are trademarks of their respective companies.

Important:
Prima Publishing has made every effort to determine that the information contained in this book is accurate. However, the publisher makes no warranty, either expressed or implied, as to the accuracy, effectiveness, or completeness of the material in this book; nor does the publisher assume liability for damages, either incidental or consequential, that may result from using the information in this book. The publisher cannot provide information regarding game play, hints and strategies, or problems with hardware or software. Questions should be directed to the
support numbers provided by the game and device manufacturers in their documentation. Some game tricks require precise timing and may require repeated attempts before the desired result is achieved.

ISBN: 7615-1810-X
Library of Congress Catalog Card Number: 98-67345
Printed in the United States of America

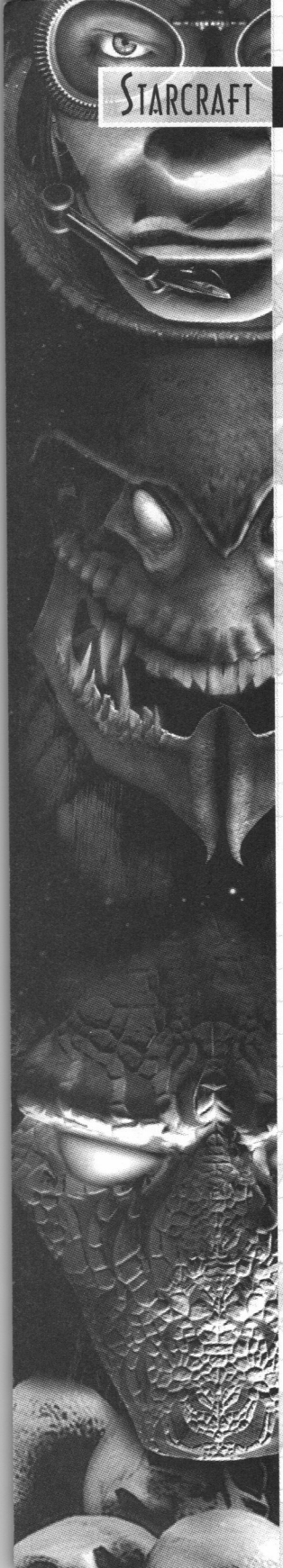
98 99 00 01 BB 10 9 8 7 6 5 4 3 2 1



Campaign Editor
Prima's Official Strategy Guide

Steve Honeywell
Joe Grant Bell

Prima Publishing
Rocklin, California
(916) 632-4400
www.primagames.com



Acknowledgments

A book of this nature requires the assistance of many people besides the author. So, to the following people I offer my sincerest thanks for their help and understanding. To my wife, Sue, who always tried to appear interested every time I invented a new Mission Briefing; to Joe Grant Bell, who laid much of the groundwork for this book; to Amy Raynor for sending projects as enjoyable as this one in my direction; to my editor Sara Wilson and the other folks at Prima who corrected my text and turned a sometimes confusing manuscript into a readable book; to my fellow students at USA Karate for stress relief; to Luna the wonderdog, guardian of the office; and finally to Gail, whose frequent Conditions of needing to be fed cause my equally frequent Actions of heating up bottles.

Contents

Chapter 1: StarEdit Fundamentals 1
- What is StarEdit 2
- What Mission Aspects Can StarEdit Affect? 3
  - Map Editing 3
  - Scenario Editing 4
  - Briefing/Linking 4
  - Triggers 5
- Limitations 6
  - Artificial Intelligence 6
  - Art and Cinematics 7
  - Miscellaneous Niceties 8
- Important Tips 8
  - Map Settings vs. Melee Games 8
  - Common Mission Types 11
- Onward! 13

Chapter 2: Interface 14
- Toolbars and Palettes 16
  - Main Toolbar 16
  - Fog of War 19
  - Sidebar 24
  - Brush Palette 26
  - Doodad Palette 27
- Pull-Down Menus 28
  - File Menu 28
  - Edit Menu 28
  - Layer Menu 29
  - Player Menu 29
  - Scenario Menu 30
  - Window Menu 35
  - Help Menu 36

Chapter 3: Building a Melee Map 37
The Groundwork 38
Making it Better 42
Base Defenses 42
Doodads 45
The Central Prize 46
Outlying Fields 48
Making it Pretty 49
Playtesting 51
Other Possibilities 51
Map Size 51
Extra Players 52
Resource Distribution 53
Complicated Terrain 53
Asymmetrical Maps 53

Chapter 4: Editing Your Scenario 56
Using Your Melee Map 58
Options with Player Settings 58
Properties Menu 58
Units Menu 61
Upgrades Menu 62
Special Abilities Menu 64
Scenario Options 66
Forces Options 66
Playing with Settings 66
A Quick Playtest 70

Chapter 5: Creating Mission Briefings and Campaigns 71
Creating Mission Briefings 71
Determining the Goal of Your Scenario 72
Creating the Mission Briefing 73
Sample Missions Briefings: Simple 80
Sample Missions Briefings: A More Complicated Example 83
Linking Missions to Make Campaigns 86

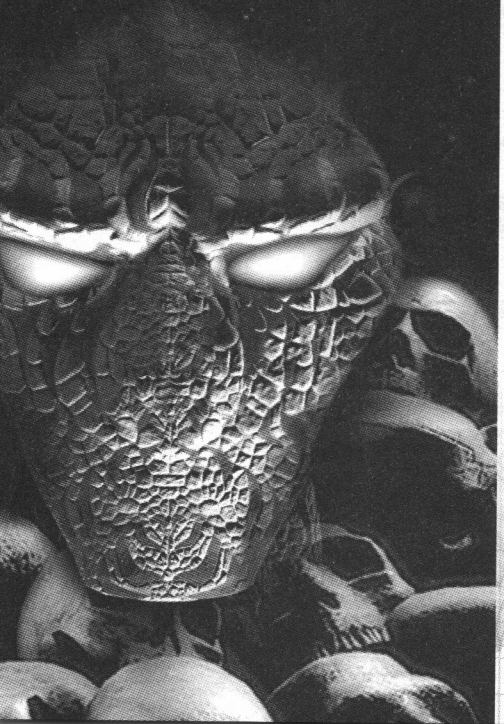
Chapter 6: Trigger Basics **89**
Definitions 90
Conditions 91
Resource Conditions 91
Always/Never Conditions 92
Bring Conditions 93
Command Conditions 93
Time Conditions 94
Kills Conditions 95
Score Conditions 96
Switches 97
Trigger Actions 97
Mission Actions 97
Victory/Defeat Actions 98
AI Actions 99
Setting Switches 100
View Actions 101
Create Actions 102
Text and Sound Actions 102
Unit Actions 104
Leader Board Actions 105
Pause Actions 107
Preserve Trigger Actions 108
Setting Countdown Timers 108
Sample Triggers 108

Chapter 7: Using Triggers to Create Unique Scenarios **112**
Using Triggers 114
Victory Triggers 114
Secondary Triggers 115
Tertiary Triggers 118
Single-Player vs. Multiplayer 118
Single-Player Triggers 119
Multiplayer Triggers 120

Unique Trigger Possibilities 120
Alarms 120
Setting Traps 121
Map Revealing 122
Taking Over Units and Structures 123
Preventing Rushes 124
Preventing "Cheats" 124
Trading Resources Among Players 125
Pitting Computer Opponents Against Each Other 126
Two-Way Teleporters 126
Unusual Scenarios 128
RPGs 129

Chapter 8: StarEdit Extra **133**
Custom AI 134
Campaign AIs 134
Choosing the Correct AI 135
Playing with Sound 135
Creating Random Events 137
What to Do Next 140
Appendix A: Keyboard Shortcuts 141
Appendix B: Unit Data 143
Structure Data 144

Chapter 1 StarEdit Fundamentals



Before you start creating maps with StarEdit, you need a firm grasp of how it works and what it can accomplish. Without this understanding, you might embark on an ambitious design only to have it fail because StarEdit doesn't work the way you thought it did.

You may be tempted to skip this chapter and get right down to the business of building maps, but keep reading: you'll pick up a lot of vital tips that will help you out down the line.

What is StarEdit?

StarEdit is a powerful editor that allows you to modify most aspects of a *Starcraft*™ mission and build new ones from scratch. You can make completely typical missions (a straightforward fight between the Terrans and Zerg, for example), or highly unusual ones (a bizarre game of interstellar football).

The missions you create with StarEdit always will have a little *Starcraft* in them. Regardless of how strangely you design your missions, they'll always retain *Starcraft*'s graphics, controls, and overall game engine. But this is to be expected—the only way to make a completely original game, after all, is to program it from scratch. That's a lot of work! The beauty of StarEdit is that it lets you work on the fun part—the actual scenario design—without doing all that programming.

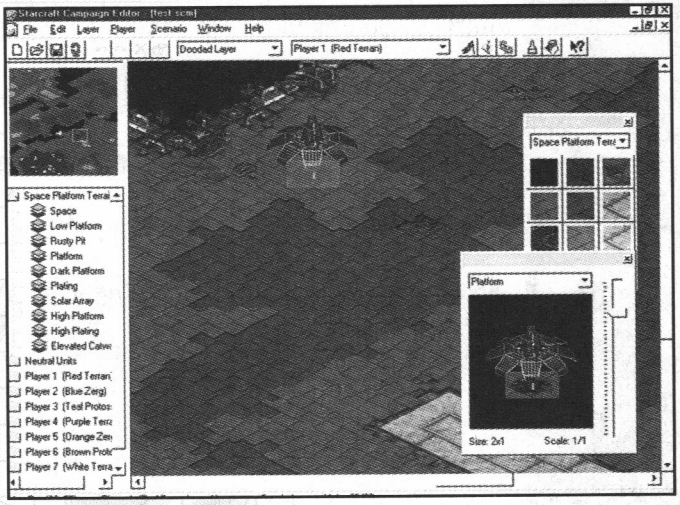


Fig. 1-1. The StarEdit program allows you to develop every aspect of your own scenarios and campaigns.

What Mission Aspects Can StarEdit Affect?

StarEdit gives you the power to change four major parts of a mission. We'll call them Map Editing, Scenario Editing, Briefing/Linking, and Triggers.

Here's an in-depth look at all four.

Map Editing

The most basic, obvious function of StarEdit is map editing. The minute you boot up StarEdit and look at the command palette, you'll see tools that let you change a map's terrain quickly and easily. Put a building here, a lake there—it's easy. Just select the appropriate terrain type and "paint" it in.

As you paint the terrain, you can add other features. These might include units and structures that human or computer-controlled player(s) can use. Perhaps you want a huge strike force of computer-controlled Protoss to lurk in one corner of the map. Go ahead and place it there, unit by unit. Or maybe you want to ensure that the Terran player gets a quick start. Easy! Just place a few Terran units and structures on the map. Each unit or structure you place is an asset the Terran gets for free at the mission's start. Now he or she will get into high gear quickly.

Chapters 3 and 4 cover map editing, but of all the things you can manipulate with StarEdit, map editing requires the least explanation. StarEdit makes it extremely easy to build a map to your personal taste.

Tip

The terms "mission" and "scenario" are used interchangeably in this book. Both terms refer to a single map designed with StarEdit.

"Campaign" refers to a series of scenarios or maps linked so the player proceeds automatically from one to the next.

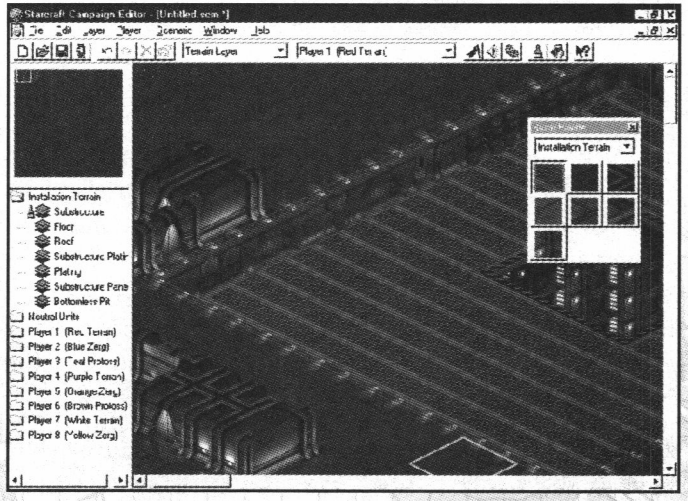


Fig. 1-2. Starting with a blank palette, you can tailor the maps you create to your specific goals and tastes.

Scenario Editing

In addition to editing the map and everything on it, you also can change aspects of the scenario. These include the following:

- ▼ Number of players
- ▼ Player species, and whether players are human or computer-controlled
- ▼ Which units, structures, upgrades and special abilities can be used by each player, and which can't
- ▼ How much it costs to research or upgrade a special ability
- ▼ Individual unit statistics, such as the amount of damage a Terran Marine deals when it attacks

There are even more, and we'll discuss them all later in Chapter 4.

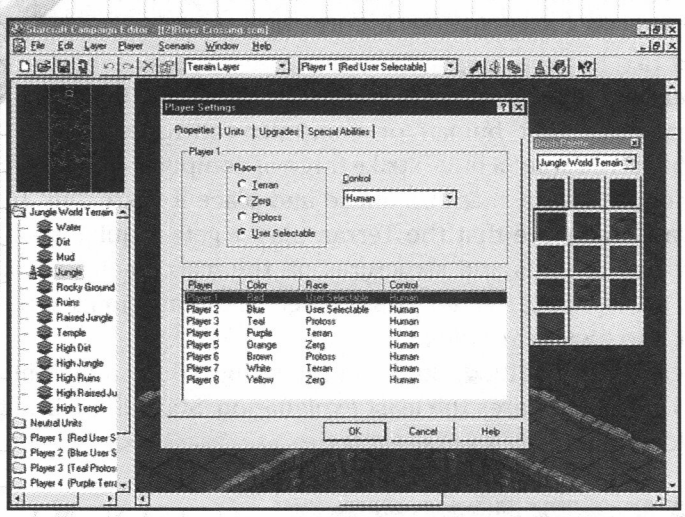


Fig. 1-3. You can control every aspect of all sides in your missions, including which side or sides will be computer-controlled and the extent to which they can develop their research tree.

Briefing/Linking

With StarEdit, you're free to edit and manipulate both the pre-mission briefing and the messages the player receives during the mission itself. You also can link scenarios into a campaign so that when he or she completes one mission, the player moves to the next automatically.

Such abilities may seem of little significance compared to changing a mission's substance, but don't underestimate them. Keeping a player informed is crucial; without a thorough briefing, the player won't know what to look for and may miss some of your clever innovations.

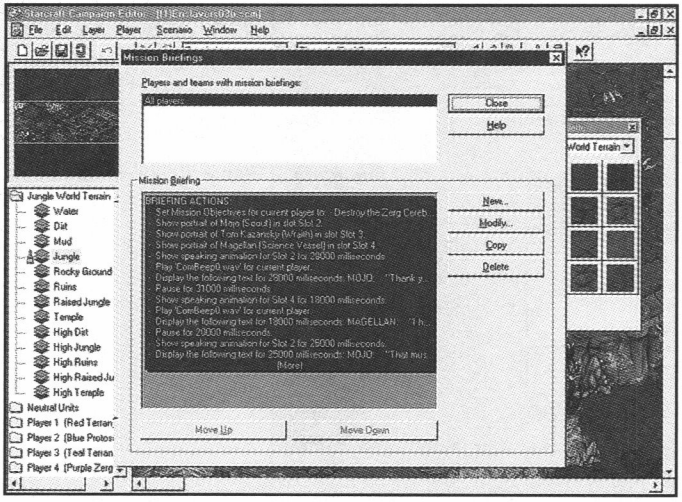


Fig. 1-4. Your mission briefings can give as much or as little information as you desire.

Triggers

Triggers, the final element you can manipulate with StarEdit, are the most powerful and complicated element of a *Starcraft* mission.

Essentially, a Trigger is just a statement that tells the mission how to behave. It consists of a *condition* that tells the mission when to activate the Trigger, and an *action* that tells the mission what to do when the conditions are met.

When you start building a new mission, you'll find three Triggers built in from the start. The third one is a good example:

Condition: Non Allied Victory players command 0 or fewer buildings.
Action: End scenario in defeat for current player.

This means that whenever a player destroys all his enemies' buildings (giving them 0 buildings to command), the scenario will end, and that player wins.

You can change the game's complexion by adding or removing Triggers. For example, removing this Trigger drastically changes the game. When it's gone, destroying all your enemy's structures no longer results in victory. So what *does* result in victory? You can set a new victory condition by adding another Trigger, of course!

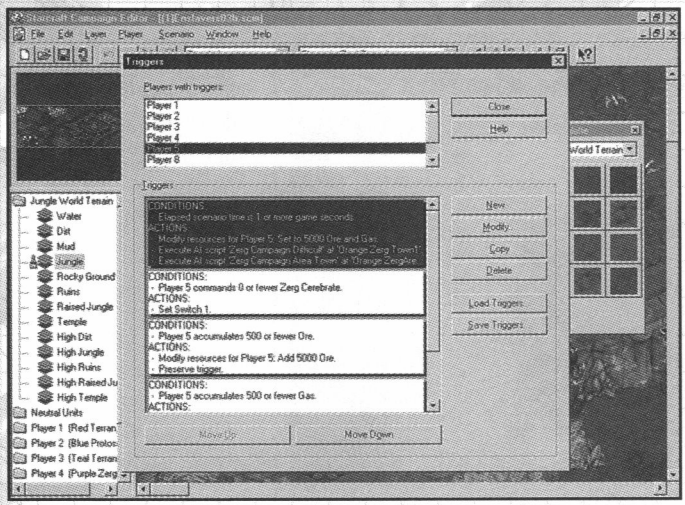


Fig. 1-5. Triggers are the most important, and potentially the most confusing, aspect of StarEdit. You'll find them covered in full detail in later chapters.

Limitations

Like every game-editing tool, StarEdit has strengths and weaknesses. You'll find some tasks easy to accomplish, and others either very difficult or impossible. This isn't because of any inherent flaw in StarEdit. All editing tools have limitations, and much of the fun of creating games lies in defeating or working around them.

This section will familiarize you with StarEdit's limitations so you don't bang your head against the wall trying to make it do something it can't. Some limitations arise from the simple fact that *Starcraft* isn't representative of every type of game. You can't make a first-person shooter or a fighter-jet simulation with StarEdit; the engine simply isn't designed for those things.

A few limitations exist because StarEdit, although similar to the editor the folks at Blizzard used to create the retail version of *Starcraft*, isn't *identical* to it. There are good reasons for this, as you'll see in the following sections.

Artificial Intelligence

Starcraft's designers used custom-built AI (Artificial Intelligence) scripts for every campaign level. These scripts tailored enemy species behavior to specific circumstances (map, resources, available units) of each mission. The designers did this to ensure that the enemy behaved as it should during each

mission: it wouldn't do to have the bad guys expanding their base when they should be attacking, or vice versa.

But this luxury isn't available to you. Creating your own AI scripts is far too complicated to accomplish using StarEdit, and the custom AI in the campaign missions is tailored so precisely to each mission, it's useless in other missions.

However, you may choose from a number of "pre-assembled" AI scripts to assign to computer-controlled forces. And you can mix and match them in conjunction with Triggers to get the computer to do more or less what you want it to do.

AI is discussed at length in Chapter 6.

Art and Cinematics

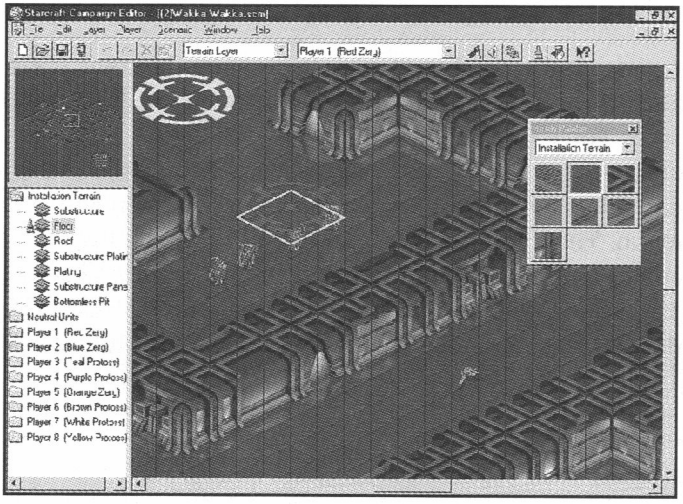


Fig. 1-6. The game's art and cinematics are the only ones you can use, but they should be enough for your needs.

I know, I know—you're just dying to replace the Terran Firebat's portrait with a picture of your cat. But you can't. Sorry.

Art is hard-coded into *Starcraft*. You can't alter it without using complex, sophisticated programming techniques. This means you can't replace unit portraits, pre-mission "splash" screens, unit animations, or in-game cinematics. Instead, you'll recycle existing *Starcraft* graphics in unique ways: what you can accomplish will surprise you.

Miscellaneous Niceties

Unless you're very wealthy, you probably don't have the resources and programmers that Blizzard has, so don't expect to modify the game in any way that isn't allowed by StarEdit. Those are the breaks. But remember, working within limitations is something everyone, even Blizzard game designers, must do.

Important Tips

The foregoing sections highlighted the things you can and can't do with StarEdit. Now you're almost ready to begin creating maps. Let's touch on just a few more issues, and then it's off to Chapter 2.

Map Settings vs. Melee Games

Those who play your mission will be able to play it in many different ways, depending on the options they choose when they start the game. It's important to understand this, so you can plan accordingly.

To play your mission, you first must start *Starcraft*, and then either select Play Custom from the single-player menu, or choose to create a new game from the multiplayer menu. Then you'll select your scenario from a list.

Now you may choose Game Type. In multiplayer games, you may select one of many options, including Use Map Settings, Melee, and Free for All. The choice between Use Map Settings and all the other options is absolutely crucial! Here's why:

Use Map Settings

When you select Use Map Settings, you're saying, "Play the mission *exactly* as the mission designer meant it to be played." This preserves everything you put into the mission:

- Every contour of the map remains the same.
- All the units and structures you placed on the map are used.
- Any global changes you made, such as adjustments to a unit's attack values or toughness, are used.
- All your Triggers are used.
- Any other changes you made to any aspect of the mission are used.

(Note that computer-controlled players will defend themselves if attacked, but they won't build or attack unless you told them to with an AI script.)

Selecting Use Map Settings allows you to play the mission exactly as it was designed: nothing added, nothing removed. This may cause problems if you failed to do something important. For example, if you erased the mission's default victory conditions but forgot to replace them with new victory conditions, you'll find that neither side can win! If you failed to assign AI scripts to computer-controlled forces, they'll just stand around with nothing to do.

Melee, Free for All, and All the Rest

When you play a mission on Melee, Free for All, or any setting that *isn't* Use Map Settings, you're telling *Starcraft* "Let's discard a lot of the unique stuff the designer did, and just use the map." That is:

- The game will use your map terrain, your resources, such as minerals and gas, and your player starting points. It also will use your Critters and Doodads (more on these later). However, the game will ignore any units or structures you placed on the map, and everyone starts the game with a standard set of units.
- The game ignores player settings from the mission (such as species and the units a player can build); players can build anything the game type they select allows.
- The game ignores all your Triggers and replaces them with generic Triggers appropriate to that game type. For example, let's say you have a mission where the goal is to capture more ore than your opponent. However, you chose to play it on Melee instead of Use Map Settings. The "get more ore" victory condition is dropped, and instead, victory is decided by Melee rules.
- The entire map starts out under a fog of war.
- Global changes to unit attributes, etc. are ignored in favor of default values.
- Computer-controlled players actively build a base and defend themselves according to the "Custom" AI script (a fairly standard script).

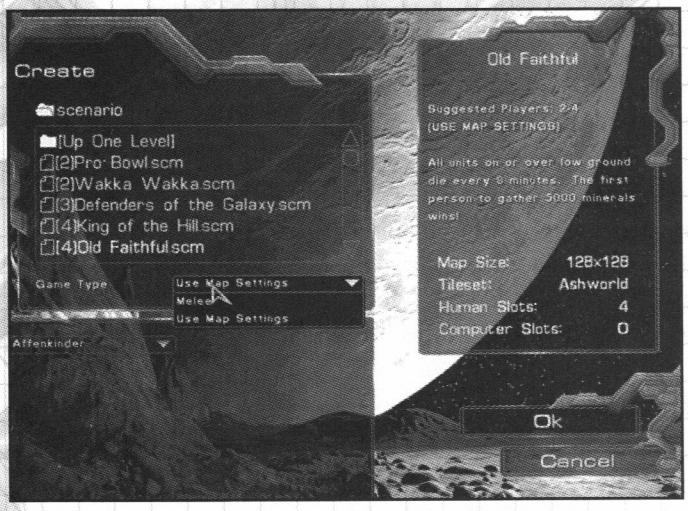


Fig. 1-7. Playing with the custom settings allows you to enjoy the scenarios' more interesting aspects.

The bottom line is that playing with anything enabled other than Use Map Settings forces the computer to ignore most of the changes you can make with StarEdit. You'll find yourself playing on a custom map, but all the other custom features you put into your mission will be ignored in favor of default settings.

Which to Choose?

When should you play a mission with Use Map Settings enabled?

If you just want to play a standard game of Melee or Free for All on a unique map, choose those game settings. And if you didn't put unique Triggers in your mission and didn't make far-reaching modifications to unit attributes, special powers, and the like, then Melee or Free for All is a great choice. You can make a mission in minutes if you plan only to use it in Melee or other standard games. In this case, you needn't worry about providing computer-controlled enemies with units, structures, or AI. Just plop down a Start Location for at least two players, adjust the map terrain until it looks the way you want it to, and save it. Voila! A new map to fight on.

On the other hand, if you build a mission with unique elements—unusual victory conditions, customized Triggers, modified unit attributes, *anything*—you *must* play it in Use Map Settings mode, or you'll lose those unique features.

In fact, you should put the words "Use Map Settings" in the first line of your description of highly customized scenarios, so players will know not to play them in Melee mode.

Common Mission Types

Although you shouldn't feel confined to building any particular type of mission, here's a brief look at the most commonly built mission types.

The Melee Map

The most basic mission you can build is simply a map for melee and other multiplayer games. These maps are meant to be played in Melee, Free for All, or any mode that *isn't* Use Map Settings.

Such missions are very simple because you build only terrain, resources, and player starting locations. You needn't worry about AI or starting forces, because Melee and Free for All modes provide each player with a standard starting force and effective AI (in the case of computer players, of course).

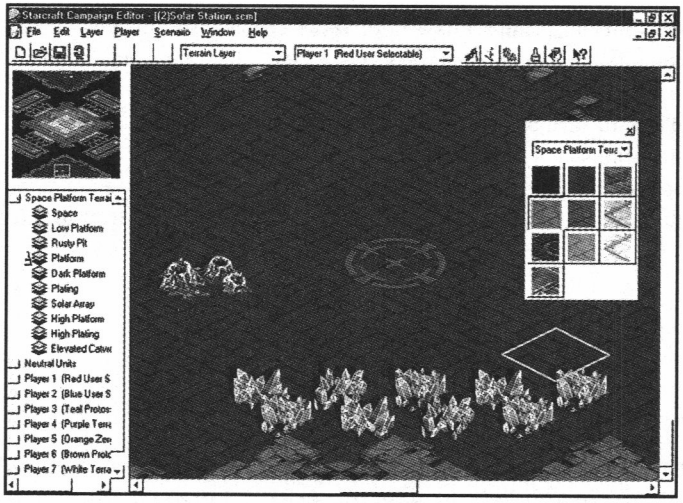


Fig. 1-8. Melee maps are designed for multiple human players to combat each other online. The two starting positions should be equal.

Your first few missions probably should be melee maps. Building these mission types familiarizes you with the process of building fair and playable maps, complete with resources, critters, and an equal opportunity for all parties to win.

You'll find it interesting figuring out how terrain and resources affect gameplay. For example, placing a vast pool of resources in the middle of the map will encourage early conflict; placing resources near each player's starting point will encourage early expansion. A map that separates players with

water, mountains, or vast distances is likely to take longer to play because the terrain will foil early attacks. Placing starting points very close to one another, on the other hand, virtually ensures a quick and brutal fight.

Combat with a Twist

After building a few standard melee maps, consider experimenting with more advanced combat missions. Advanced missions almost always include some sort of customization, so they must be played with Use Map Settings enabled. Of course, this means you must add units and AI for computer-controlled enemies because *Starcraft* won't give them a starting force and sense of purpose automatically.

Any number of ways exist to take a basic combat map and change it radically using a few quick modifications. Here are a few examples of quick, easy changes.

- Add a large, powerful force to the map and set its control to "Rescuable." Now the first player to reach it will assume control of that force. Now you've got a game where early exploration is extremely important.
- Limit the unit types available to each player, and see how this affects the game.
- Adjust unit abilities or the cost of special attacks.
- Add "wild card" computer-controlled forces to complicate the battlefield.
- Adjust the victory conditions using Triggers, so players must do more than just fight to win.

More possibilities are explored in Chapter 7.

Unusual Missions

The next step, should you want to take it, is to create missions where combat isn't the principal goal. For example, let's say you create a mission that gives each player five identical units and places them on a battlefield covered with 10 flags. Using Triggers, you can adjust victory conditions so the player who touches all 10 flags first wins. The result is a game that allows two general paths to victory. Players can have their units fight it out, with the winner's units going on to touch the flags and win, or they can try to avoid combat and touch all the flags without getting embroiled in a fight.

Thus, it's possible to create *Starcraft* missions that bear very little resemblance to a "standard" mission.

Complete Overhauls (For Example, RPGs)

Finally, you can customize missions until they barely fit the mold of a typical real-time strategy game. You'll find examples of this in your copy of *Starcraft*. Look in the maps/scenario subdirectory of the *Starcraft* directory, and you'll find Wakka Wakka, a clever remake of an old arcade classic. Another example of a truly unique mission is Pro Bowl, which simulates a violent game of interstellar football.

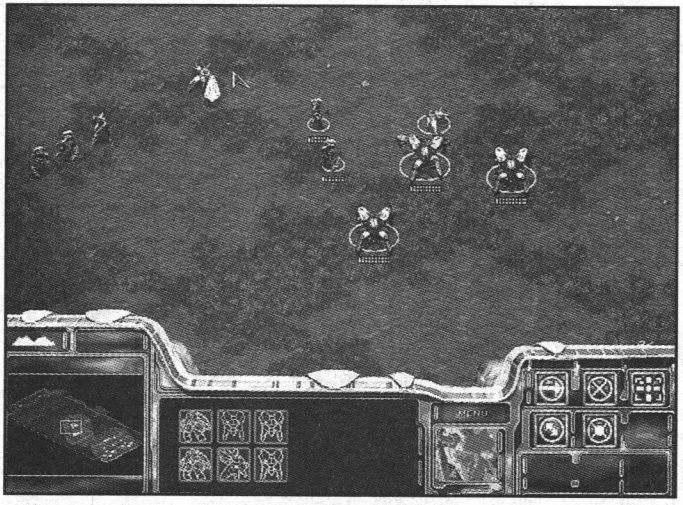


Fig. 1-9. In the Pro Bowl scenario, you aren't trying to kill the enemy, but to score touchdowns!

Many players designing *Starcraft* missions try to create RPGs (role-playing games). In such missions, the player (or players) control a single unit apiece and travel through mazelike maps. When their unit kills enemies, it gains "experience points" as in a traditional RPG, and eventually gains levels based on this experience. These games usually also feature a rudimentary system of inventory, so a player's lone unit can "pick up" items and use them later.

The RPG is a difficult mission type to implement because it forces the *Starcraft* engine to do things it was never meant to do, such as keep track of inventory. Later chapters explore techniques for making RPGs.

Onward!

You know the basics. Now it's time to get your feet wet by actually building missions. The next chapter focuses on using StarEdit's map tools to create a playable Melee map, a simple map type.

Chapter 2
Interface

Before diving right in to build maps, let's quickly tour StarEdit's interface. It's fairly easy to figure out. You may be daunted by the range of options at first, but it's soon clear the basic commands are organized logically and accessibly. Furthermore, even though StarEdit has lots of toolbars, palettes, and pull-down menus, there aren't really that many commands to master. There's a lot of overlap in those toolbars, palettes, and menus—that is, different buttons or menu options execute the same commands. This is all to your benefit, because you can pick and choose what works best for you.

Toolbars and Palettes

StarEdit features several toolbars and command palettes. Often, their buttons and selections duplicate commands in the pull-down menus. Most users will prefer to use toolbars and palettes when they can, because they're usually faster.

Main Toolbar

The main toolbar runs along the top of the screen, just beneath the pull-down menus. It contains a number of useful buttons and tools.

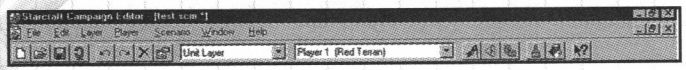


Fig. 2-1. The Main Toolbar

Standard Options Buttons (New, Open, Save)

The first three buttons on the main toolbar are New, Open, and Save. These buttons allow you to perform basic file functions.

- *New* creates a brand-new map. When you press the button, a dialog box appears that lets you specify your new map's size and basic terrain type. Be sure to save your old map before creating a new one!
- *Open* lets you open a pre-existing map. This can be a map from the *Starcraft* CD, a map you downloaded from the Internet, or a map you created earlier.
- *Save* lets you save your current map. If you've already assigned this map a name, pressing the button simply saves your work. If you haven't assigned it a name yet, a dialog box appears and prompts you for one.

Be sure to save files where you can find them later.
The New button presents you with several options that are discussed in detail in Chapter 3.

Compact Button

Press the Compact button to "squash" your map file and create a smaller file. You can't reverse this, but it doesn't matter, because you don't lose anything by compacting your map. Press this button every once in a while, and again when you're through editing your map. Small maps are easy to send via the Internet, so if your map is small it's more likely to be downloaded and played.

UNDO/REDO BUTTONS

The Undo button allows you to "take back" changes. You can press this button multiple times to take back multiple actions. For example, let's say you drop three Terran Marines on the map, but then decide you only want one. Press Undo twice to take back the last two Marines you added.

If you second-guess your Undo, press the Redo button to restore the map to its previous state. Following the Terran Marines example, you could press Redo twice to replace those two Terran Marines.

Tip

To select a unit, enter the Unit layer (see "Layer Picker") and then either click on the unit or drag a selection box around it.

CLEAR BUTTON

Select any number of units or Doodads, and then press the Clear button to remove them from the map.
Pressing the Delete key on your keyboard achieves the same effect.

PROPERTIES BUTTON

Select a single unit or structure, or multiple units or structures, and then press the Properties button. The Properties menu appears.

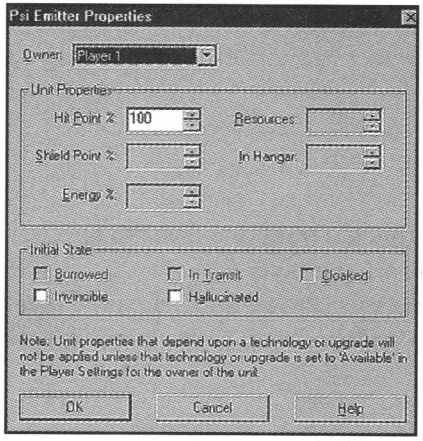


Fig. 2-2. Use the Properties menu to change the beginning properties of any unit, structure, resource, or special item.

Tip

In addition to combat units, you also can select neutral units, such as Critters (Rhynadons, Ragnasaurs) and resources (Vespene Geysers, Mineral Fields).

As figure 2-2 shows, the Properties menu allows you to adjust both the selected units' properties and initial state.

- *Hit Point %* allows you to lower the unit's Hit Points (health) to a percentage of its normal value, in order to create units that are injured or below average health at the mission's start.
- *Shield Point %* and *Energy Point %* allow you to lower units' starting Shield and Energy values to a percentage of normal. Obviously, if the unit possesses neither shields nor energy, you can't adjust these.
- *Resources* allows you to adjust the total amount of Vespene or crystals in a selected Vespene Geyser or Mineral Field.
- *In Hangar* is available only for Protoss Carrier and Reaver units. This value adjusts the number of Interceptors or Scarabs that start inside each unit.

Tip

To put more than four Interceptors into a Carrier, make sure the Carrier Capacity upgrade has been developed. Similarly, to place more than five Scarabs into a Reaver, be sure the Reaver upgrade has been developed.

- *Burrowed*, when checked, allows units with Burrowing ability to start the game in the burrowed state. (This option is available only to certain Zerg ground units).
- *In Transit* is an option available for Terran structures that fly. Check this box to have the structure start the game in the air.
- *Cloaked* allows units with Cloaking ability (Terran Wraith, Terran Ghost) to start the game cloaked. As with In Hangar, though, the appropriate special ability must be researched for this to work.
- *Invincible* renders a unit immune to damage. It's not only impossible to kill, but can't be targeted by enemy attacks or special abilities.
- *Hallucinated* makes a unit a Hallucination, like the High Templar spell. This limits its life span, and it takes double damage from attacks.

As you can see, not all options are available to all units. Also, some options (such as Hallucinated) are hard to use effectively.

Note that you can access the Properties menu in several ways. It appears under the Edit pull-down menu, has an easy keyboard shortcut (Enter), or you can select a unit, right-click, and then select Properties from the pop-up menu.

Layer Picker

Tip

Layers allow you to edit different parts of a map with ease. Select the Unit layer to place, remove, or edit units. Select the Terrain layer to place, edit, or alter terrain. It's simple: to edit something, first switch to the appropriate layer.

The Layer Picker lets you change quickly from one layer to another. It's easy to use—if you understand what a layer is.

To switch layers, click on the Layer Picker and select the layer you want. There are five layers to choose from; here's a quick look at all five.

- *Terrain* lets you edit the land itself. Bridges, trees, and other special items are called "Doodads," and aren't considered terrain components. Doodads get their own layer.
- *Doodad* allows you to place and edit Doodads, which are special features that you can place on top of the terrain. Most Doodads are merely decorative, although the biggest Doodads act as obstacles land units must walk around. A few Doodads, such as doors and bridges, can have great tactical significance.
- *Unit* lets you place and edit units, resources, and structures.
- *Location* lets you draw location boxes on the map. These boxes are invisible to the player, but they can be highly significant when used with Triggers. For example, with the appropriate trigger, a player might win the mission if one of his or her units enter the location box.
- The *Fog of War* Sidebar provides details on this complex layer.

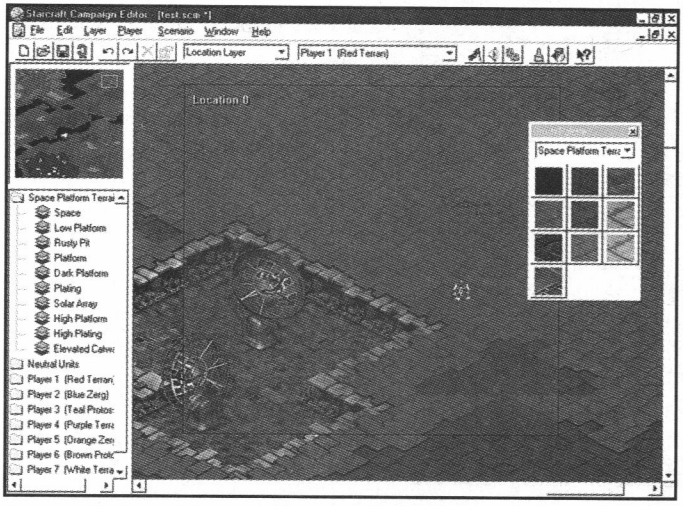


Fig. 2-3. The Location layer allows you to specify important map areas.

Fog of War

When a game begins, Fog of War typically obscures the entire map. Only by sending units to remote corners of the map can players reveal everything.

When you switch to the Fog of War layer, you can "paint" areas of the map to make them lighter. These lighter areas will be visible to a player from the start of the game. This is useful, for example, if you design a mission that requires a player to reach a certain point on the map. You could paint that spot in Fog of War view, so the player knows from the start what the destination is.

Each player has different Fog of War settings. You must select each player using the Player Picker (see below) and adjust the Fog of War to reveal what that player should see.

Also, note that, although players can see the general terrain of the revealed (or "nonfogged") area, they can't see units moving around there, unless one of their own units is nearby.

Player Picker

Just as the Layer Picker lets you select which layer to edit, the Player Picker lets you choose which player's forces to edit.

Select a player from the Player Picker. That player is now active. The Brush Palette now lets you place units that belong to the selected player, and the Player Settings menu lets you adjust settings for the chosen player. This is explained in detail in Chapter 4.

Forces Button

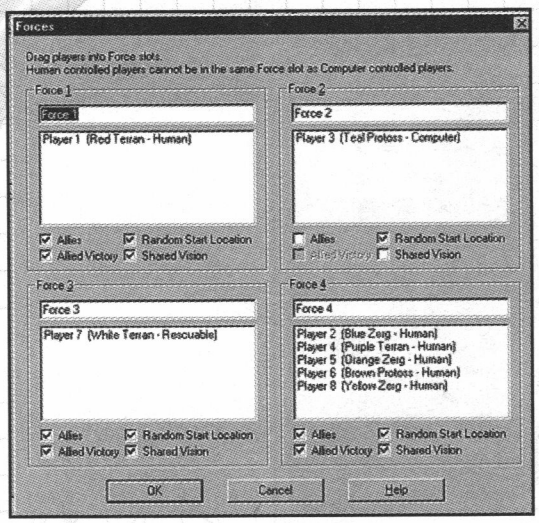


Fig. 2-4. The Forces menu allows you to organize the different armies into up to four sides.

The Forces button brings up the Forces menu (figure 2-4). This menu provides an easy way to organize players within a scenario.

Click and drag players to move them from one Force slot to another. Then adjust several checkboxes to determine how players on the same force interact:

Allies: Players on a force together start the game as allies. Their units won't attack one another.

Allied Victory: If one player on a given force meets the victory conditions (established with Triggers), all players on that force share the win.

Random Start Location: Assigns players within a given Force a random Start location.

Shared Vision: Allows players to see what other players in the same force can see. For example, if Player 1 sends a scout into uncharted territory, Player 2 (on the same force as Player 1) also can benefit from the scout.

Although it's a good idea to separate opposing players onto different forces, it isn't always necessary. Melee maps ignore the Force Settings, for example.

Human players can't share a Force slot with computer-controlled players.

Once you've assigned players to different forces, you can create Triggers that affect only one force, customizing the game for each.

Finally, note that players on the same force can adjust the Allies, Allied Victory, and Shared Vision settings once they're in the game itself. These options are available from the in-game Diplomatic Settings menu. Thus, teammates can part ways whenever they choose.

Tip

You must use the Opponents condition in your Victory Triggers to make an Allied Victory work. This is discussed in detail in chapters 4, 5, and 6.

SOUNDS BUTTON

Press the Sounds button to bring up the Sound File Manager (figure 2-5. This allows you to import .wav files into StarEdit and use them as sound effects, narration, and character dialogue.)

To import a sound, access the Sound File Manager and press the Import button. Locate a .wav file and select it. It will appear in the main window of the Sound File Manager, along with three types of information:

Seconds, the length of the sound file;
Triggers, the number of times the sound is used in a Trigger action;
Briefings, the number of times the sound is used in a Mission Briefing action.

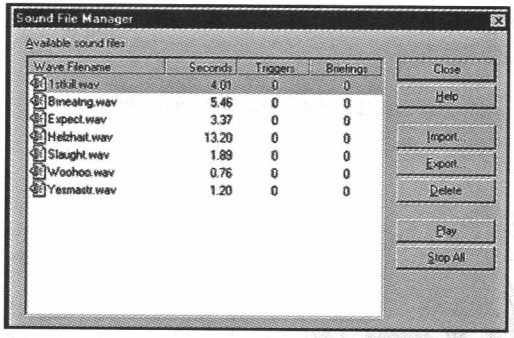


Fig. 2-5. Import .wav files to customize your missions to a much higher degree.

Tip

Where can you get .wav files?

- Download them from the Internet.
- Record them with a utility such as the Sound Recorder included with *Windows 95* and *98*.
- Unpack them from *Starcraft* and insert them in new places. The tool used to do this is discussed in Chapter 8.

The buttons alongside the Sound File Manager have the following functions:

Close closes the Sound File Manager.
Help brings up information about the Sound File Manager.
Import imports new .wav files.
Export allows you to save a selected .wav file anywhere on your computer, under any name you choose.
Delete removes the sound from the map and disables any Triggers that use it.
Play plays the sound.
Stop All stops the sound playback initiated by Play. This is useful for long sound files.

Note

Because sounds work in conjunction with Triggers, I won't discuss them extensively until Chapter 6.

Although the Import button imports a sound into the map, it can't put the sound to use. Triggers—specifically, those labeled "Play Wav" and "Transmission"—initiate a sound during a mission briefing or during the game. The Trigger function is discussed in detail in chapters 5 and 6 of this book.

Triggers Button

Press the Triggers button to access the Triggers menu (figure 2-6).

Triggers are powerful scripting tools that allow you to modify the mission briefing, the victory conditions, and everything in-between. Learning to use Triggers is both fun and crucial. Much of the latter part of this book is devoted to them.

The Triggers menu contains several elements. Near the top is a pane labeled "Players with Triggers." Click on any player listed in this pane, and a list of Triggers applying to that player will appear in the large pane below it.

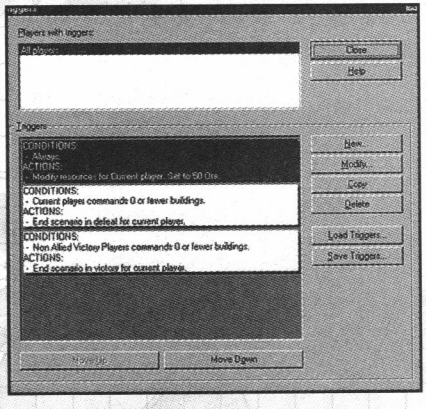


Fig. 2-6. Learning the Triggers system is the most important part of developing your own unique scenarios and campaigns.

Below the Players with Triggers pane is a large pane labeled simply "Triggers." This window lists all Triggers affecting the current player, or, optionally, every Trigger in the game (if All Players is selected).

You'll notice several buttons to the right. In addition to the standard Close and Help buttons, you'll find these:

New steps you through the process of creating a new trigger.
Modify lets you modify an existing selected trigger.
Copy copies an existing selected trigger—useful in creating a new trigger almost identical to an old one. Just copy the old one and then modify it to your liking.
Delete gets rid of the selected trigger.
Load Triggers lets you load a set of triggers you saved earlier.
Save Triggers saves all triggers from this map into a file you can load later—handy for remembering how you did something in an earlier map: just Save Triggers, and load them later to look over your handiwork.

You'll spend more time working with the Triggers function as you proceed into chapters 5 through 7.

New Brush/Doodad Palette Buttons

The New Brush Palette button makes a Brush Palette appear onscreen. This is useful if you've closed the Brush Palette, and want to use it again.

Similarly, the New Doodad Palette button causes a Doodad Palette to appear.

You can have multiple Brush Palettes or Doodad Palettes on the screen at once, but because you can have only one palette selected at any given time, there's really no need.

You'll find details on Brush and Doodad palettes later in this chapter.

Help Button

The Help button displays Help files that explain key StarEdit features. Almost all this information also appears in this book, but you'll appreciate being able to search for help on specific topics as you refer to these files.

Sidebar

The Sidebar is a big pane that covers the left edge of the screen (figure 2-7). It includes a Mini Map that displays a miniature view of the entire map at once, and the Tool Tree, which looks like a big list of folders.

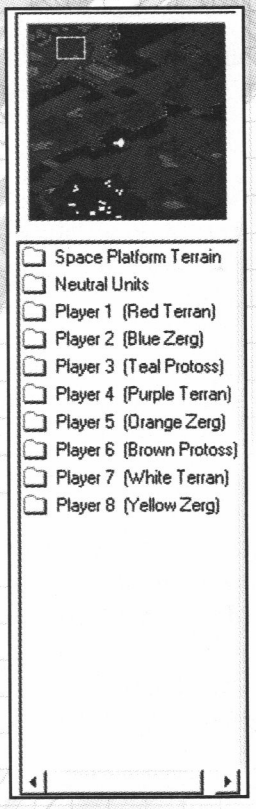


Fig. 2-7. The Sidebar allows you to move from place to place and from army to army quickly.

Mini Map

The Mini Map shows you the entire map at once. It's small, but it's invaluable when you need to see how the part of the map you're editing fits into the big picture.

Also, you can move around the map quickly by clicking and dragging in the Mini Map.

Units on the Mini Map are color-coded as follows:

Light Blue = Neutral Units (Resources, Critters, Power-Ups, and unattached Add-ons)
Red = Player 1
Dark Blue = Player 2
Teal = Player 3
Purple = Player 4
Orange = Player 5
Brown = Player 6
White = Player 7
Yellow = Player 8

Terrain Folder

The big list of folders just below the Mini Map is called the Tool Tree. The Terrain folder, at the top of the folder list, allows you to paint various types of terrain onto the map.

Click on the Terrain folder to expand it and view all possible terrain types for the current map's tileset. Then click on the terrain type you want (say, Lava) and paint it onto the map. To paint large areas, click and drag; to paint small areas, click on individual spots.

Some terrain is *impassable*—that is, you can't place units there, and players' land units can't cross it during the game. Also, some terrain is *unbuildable*: players can't build structures there. Common sense usually tells you which terrain is suitable for which tasks. Lava, for example, can't be built on.

Tip

The Terrain folder isn't labeled "Terrain Folder." Instead, it's named after the tileset the map is based on. If you choose Badlands as the tileset, for example, the Terrain folder will be labeled "Badlands Terrain."

Neutral Units Folder

The Neutral Units folder allows you to place neutral units (Critters) and resources on the map. Critters are alien life forms unowned by any player. Because they aren't hostile, their effect is mainly decorative.

Resources, on the other hand, are absolutely crucial to the game. Place Vespene Geysers and Mineral Fields carefully. Then select them and press the Properties button to edit the amount of resources each Geyser or Mineral Field bestows.

Tip

Selecting anything from the Terrain folder automatically puts you in the Terrain layer, so you can start modifying the landscape immediately.

Tip

Doodads and Units are removed from the map automatically if you change the terrain beneath them to a type that can't support them.

Player Folders (1–8)

Rounding out the Tool Tree are eight folders, devoted to one player each. Select a player's folder to access subfolders; then select individual units and place them on the map.

Here's a look at those subfolders:

Ground Units contains ground units, such as the Terran Marine.
Air Units contains airborne units, such as the Terran Wraith.
Heroes contains special units that possess superior abilities.
Special contains special units such as Map Revealers (invisible units that reveal parts of the map to a player whose units stumble across them) and Flags (these can be important, depending on the Triggers you use with them).
Buildings contains structures.
Addons contains Add-ons, such as the Comsat Station, which attaches to the Terran Command Center.

Tip

The three Mineral Fields listed in the Neutral Units folder are identical except for appearance.

Tip

Add-ons can be placed alone, without their "host" buildings. To take advantage of them, construct a host building that connects to the Add-on.

Special Buildings contains special structures that aren't needed to produce units, but which can serve as decorations or mission goals.
Start Location contains the player's Start location, and that determines where the player's view is centered when the mission starts.
Power Ups contains Power-Up objects that perform various functions typically involving a boost of skills to the unit(s) that found them. (This is done with Triggers.)

Brush Palette

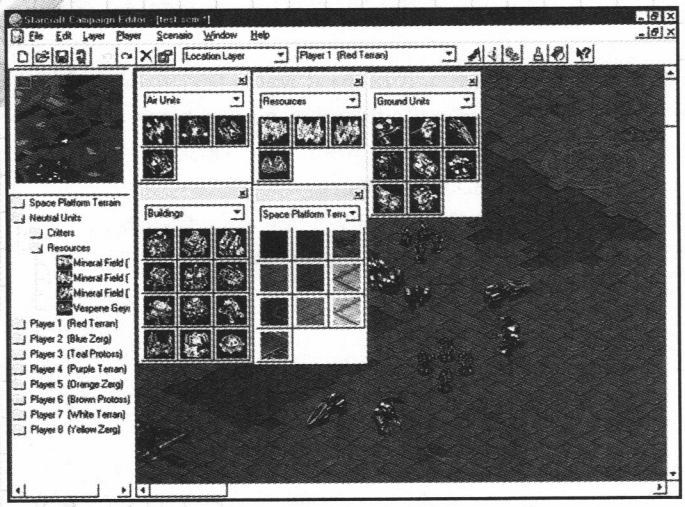


Fig. 2-8. Although the Brush Palette duplicates the Tool Tree in the Sidebar, often it's more efficient.

Tip

To place player-specific units with the Brush Palette, first select the player with the Player Picker. Otherwise, you'll find yourself placing combat units or structures belonging to someone other than the player you'd intended.

The Brush Palette (figure 2-8) floats on the screen. You can move it to wherever you'd like it, or close it. Press the New Brush Palette button on the Main Toolbar to create a new Brush Palette, if you closed the old one.

The Brush Palette's function is almost identical to the Tool Tree's. As with the Tool Tree, you'll use the Brush Palette to select terrain, units, and structures so you can paint them onto the map.

Whereas the Tool Tree uses folders, the Brush Palette has a pull-down menu for selecting categories. Also, items in the Brush Palette display as icons instead of text descriptions. Finally, you can move the Brush Palette to wherever you want, unlike the Tool Tree, which you can resize but never move.

So, which should you use, the Tool Tree or the Brush Palette? It's up to you. Use the tool you find most convenient for the job at hand.

Doodad Palette

The Doodad Palette (figure 2-9) allows you to place Doodads on the map. Again, Doodads are features you can place on the terrain. Most Doodads are simply decorative, but because most of them will block the movement of land units, you can use them as obstacles.

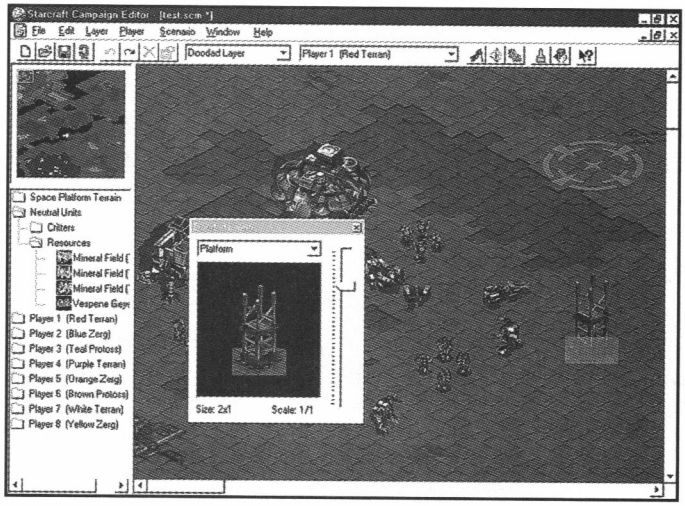


Fig. 2-9. The Doodad Palette allows you to place scenery and obstacles on the map. Each terrain type has specific Doodads that go with it.

Tip

A few Doodads serve special purposes, and are significant in ways other Doodads aren't. Bridges, stairs, and ramps, for example, are Doodads that allow ground units to travel between previously unconnected map areas. Placing these Doodads can be tricky, because they must sit on the border between the areas they connect.

Every Doodad belongs to a particular type of terrain. Click on the pull-down menu at the top of the palette to select a terrain type. Then use the slider bar to the right to scroll through the various Doodads. Click on the picture of the Doodad itself to select it as your brush; then paint it on the map.

Selecting a Doodad puts you in the Doodad layer automatically.

You can place Doodads only on their native terrain, and you must place some Doodads, such as ramps and bridges, very carefully. When you try to place a Doodad, the color of the cursor indicates where you may put it. A fully green cursor tells you that you can place the Doodad, fully red means you can't place a Doodad, and part

Note

The Doodad Palette mentions certain types of terrain that don't appear in the Brush Palette—Cliffs, for example, on the Badlands tileset. These "mystery" terrain types are generated automatically as transitions between the terrain types you *can* place directly. For example, Cliffs are created when you place High Dirt next to regular Dirt.

Tip

When placing Doodads that connect high and low ground, or one piece of land to another (ramps, stairs, bridges), make the border between the two terrains as straight as you can. Ragged dividing lines rarely provide room enough to place Doodads.

green, part red means that part of the area can sustain the Doodad, but the other part can't. In that case, you either should move the cursor until it's fully green, or adjust the terrain until it can support the Doodad.

Pull-Down Menus

Seven pull-down menus—File, Edit, Layer, Player, Scenario, Window, and Help—appear across the top of the screen. Most of the options in these menus also appear on various toolbars and palettes, and have been discussed already.

File Menu

The File menu provides a number of standard options for opening, saving, and creating new files. You'll also find a Compact option, which shrinks the size of your map.

Standard Options (New, Open, Save, Save As)

The New, Open, Save, and Save As buttons are standard to all *Windows 95* and *Windows 98* applications. The "Main Toolbar" section earlier in this chapter discussed these functions.

Compact

The Compact button compresses your map so the file is smaller. It has no effect on gameplay.

Edit Menu

The Edit menu contains a few options available on the Main Toolbar, plus a few options available only here.

Undo/Redo

Undo allows you to take back multiple actions, and Redo lets you restore them.

CLEAR

Clear removes all selected units, structures, or Doodads from the map.

First, select the things you want to delete, either by clicking on them individually or by drawing a selection box around them. Then press Clear to remove them. Note that the Delete key achieves the same thing.

SELECT ALL

Select All selects all units or Doodads on the map. If you're currently in the Unit layer, all units are selected; if you're currently in the Doodad layer, all Doodads are selected.

SHOW GRID

Show Grid superimposes a grid over the map. Select it again to make the grid disappear.
The grid can be useful for measuring distances.

PROPERTIES

Properties brings up the Properties menu, discussed earlier.

Layer Menu

The Layer menu is virtually identical to the Layer Picker. Select Terrain, Doodad, Unit, Location, or Fog of War to move to the relevant layer.

Player Menu

Player menu commands apply to individual players, and allow you to switch from one player to another.

PLAYERS 1–8 (RED, BLUE, TEAL, PURPLE, ORANGE, BROWN, WHITE, YELLOW)

The top eight selections of the Player menu function identically to the Player Picker on the Main Toolbar. Select the desired player from among the eight players shown. Then any units or structures you place with the Brush Palette will belong to that player.

Go to Start Location

Select Go to Start Location to center your view on the current player's Start location. If the player doesn't have one yet, nothing happens.

Settings

Settings brings up the Player Settings dialog box, which allows you to adjust key features for each of the eight players. It contains the following:

Tip

Only human-controlled players can have a User Selectable species.

Player Grid: A window that shows every player, and the color, species, and control for each. Click on players in this grid to edit them.

Race Picker: A set of radio buttons that lets you select a different species for each player. Leave this as User Selectable if you want players to select their own species.

Control Picker: A pull-down menu that lets you decide if the player is human-controlled, computer-controlled, rescuable or neutral.

Note

Rescuable units don't move until a human player sends units to "rescue" them. The rescuable units then fall under the rescuing player's control. Note that, if a rescuable player's Colony Center is rescued, the rescuer gets *all* units belonging to that player.

Neutral units will do nothing until they're attacked. Then they deem their attacker an enemy and defend themselves.

Scenario Menu

The Scenario menu allows you to make major adjustments to the scenario as a whole, including edits to such major features as Force settings, Triggers, and Mission Briefings.

Forces Menu

Select Forces to bring up the Forces menu. See the "Forces Button" section earlier in this chapter for details.

Sounds Menu

Click on Sounds to summon the Sounds menu. See the "Sounds Button" section earlier in this chapter for details.

Triggers

Select Triggers to bring up the Triggers menu. See the "Triggers Button" section earlier in this chapter for details.

Mission Briefings

The Mission Briefings option brings up the Mission Briefings menu (figure 2-10). This menu allows you to create the briefing a player sees before playing your mission. You can develop most of the story for your mission at this point.

A number of options allow you to customize your mission. You'll find further discussion on the triggers that control the Mission Briefings in later chapters covering the Triggers function.

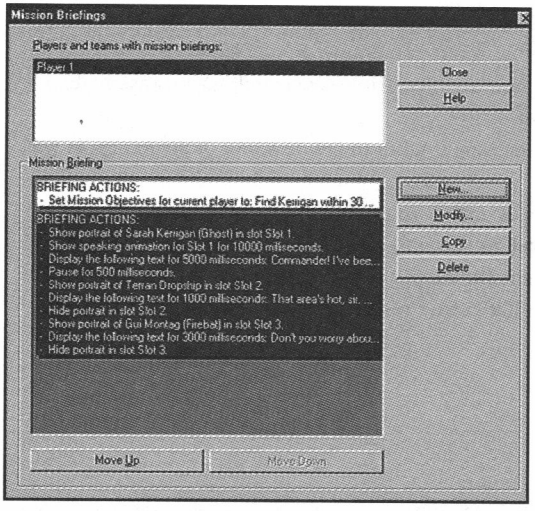


Fig. 2-10. A well-done briefing gives the player a good sense of the battle ahead.

Unit and Hero Settings

The Unit and Hero Settings option brings up the Unit and Hero Settings menu (figure 2-11). From here, you can modify the units, structures, and heroes the player and computer-controlled enemies will create. You'll keep these at standard settings most of the time, but in a few cases you may wish to augment or change the units' power. You have a lot of leeway in what you can change.

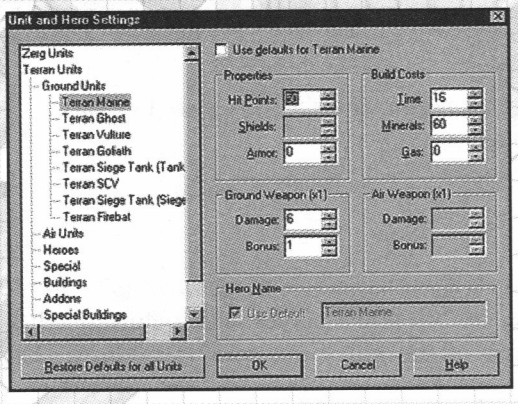


Fig. 2-11. The Unit and Hero Settings menu allows you to customize your scenario further.

Using the slider bar at the left, pick the unit or structure you wish to change. A list of options will appear:

- *Use Defaults*, when clicked off, allows you to change the settings of the unit or structure in question.
- The *Properties* box lets you change hit point number, armor level, and, for Protoss units, shield strength for each unit or structure.
- From the *Build Cost* box, you may alter the time it takes to create a unit or structure, as well as the cost in minerals and gas.
- The *Ground Weapon* and *Air Weapon* boxes enable you to change the amount of damage a unit or defensive structure causes, as well as the amount of increase for each upgrade.
- The *Hero Name* dialog box lets you change the names of your hero units.
- The *Restore Defaults* button in the lower-left corner resets any changes you make to these settings across the board.

Upgrade Settings

The Upgrade Settings option brings up the Upgrade Settings menu (figure 2-12). With this menu, you can alter the cost and time it takes to create upgrades for the mission's units and structures. As with the Unit and Heroes Settings, generally you'll leave these alone, but augmenting them will give your scenario a different feel. You have fewer options than in the Unit and Heroes Settings menu.

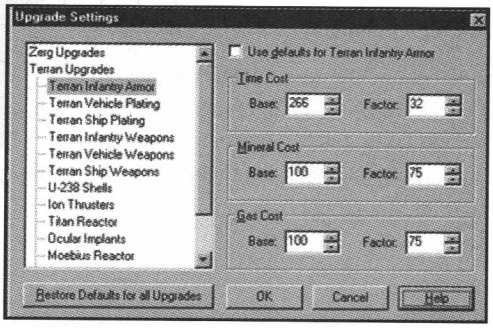


Fig. 2-12. Not happy with the way the upgrades progress? Use this menu to change them.

- ▼ *Use Defaults*, when it's clicked off, allows you to change upgrade aspects.
- ▼ The *Base* figures for time, minerals, and gas are the figures used for the first upgrade of a particular type. Look at figure 2-12: the time to create the initial upgrade to Terran infantry armor will take 266 seconds, and will cost 100 minerals and 100 gas.
- ▼ *Factor* figures are used for additional upgrades. If something can be upgraded multiple times, the second upgrade adds the factor numbers to the base cost. Again, refer to figure 2-12: the second upgrade for Terran infantry armor takes 298 seconds (266 base plus factor of 32) and costs 175 minerals and 175 gas (100 base cost plus factor of 75). A third upgrade is even more expensive, taking 330 seconds (266 base plus factor of 32 plus another factor of 32) and costing 250 minerals and 250 gas (100 base plus factor of 75 twice).
- ▼ The *Restore Defaults* button returns any upgrade changes you make on this menu to the original settings.

Special Ability Settings

The Special Ability Settings option brings up the Special Ability Settings menu (figure 2-13). This menu is similar to the Upgrade Settings menu, but it controls research cost, resource cost, and energy expense for units' and heroes' various special abilities.

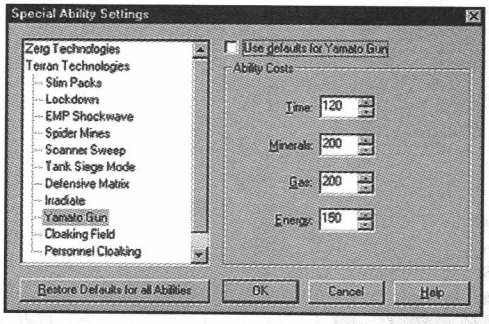


Fig. 2-13. Use this menu to change costs for unit special abilities.

- When the *Use Defaults* button is clicked off, you can change the costs of creating and using a special ability.
- The *Time*, *Mineral*, and *Gas* buttons allow you to change the time it takes to research the special ability, as well as the mineral and gas cost for developing that ability.
- The *Energy* button changes the amount of energy it takes to use the special ability.
- The *Restore Defaults* button returns all altered special ability properties to their default settings.

Properties

The Properties option brings up the Properties menu. From here, you can name your scenario and add other information a player might find useful. This information appears when the player loads a new scenario. The Description box allows you to type in a short briefing of the mission at hand and any other information you think is important. The following are among the things to consider including here:

- Use Map Settings: This gives players the specific objectives and properties changes you're including.
- Number of players: This suggests the number of players best for the scenario.
- A brief overview of the mission.

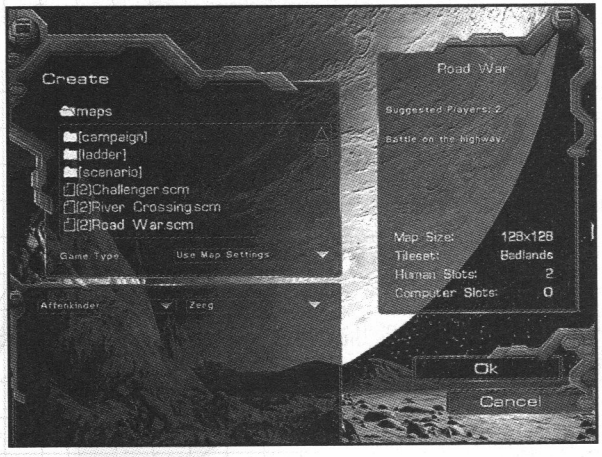


Fig. 2-14. Your mission's name and listed properties display on the right side of the screen when you choose a new scenario.

Window Menu

The Window menu allows you to call up new or additional tools and change the scenario's configuration onscreen so you can work more efficiently.

New Brush/Doodad Palette

When you enable these options, they create a new Brush Palette or Doodad Palette. Multiples of each can display onscreen at the same time.

New Map View/Close Map View/Close All Map Views

If you wish, you can have multiple versions of the map view visible at all times, allowing you to move quickly from place to place on the map. Having a map view open at each base in the scenario lets you center a view on each, for instance.

The New Map View option opens an additional map view. Close Map View shuts the currently selected view. Close All Map Views closes all open map views.

Cascade/Tile Horizontally/Tile Vertically

These options display your map views in different ways. The Cascade option places them one behind the other so that you can click on each. Tile Horizontally arranges all map views vertically, one over the other; each appears as a long horizontal strip. Tile Vertically does the opposite, placing all map views in a single row of tall, thin vertical strips.

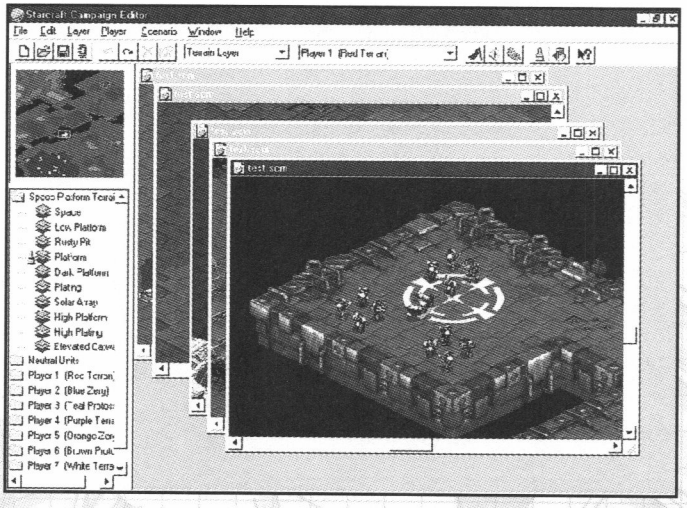


Fig. 2-15. Use the Cascade option when you have multiple views of the map open at once.

Note

Remember, scenario name and file name are two different things. The default name for a scenario is "Untitled Scenario." The default description is "Destroy all enemy buildings."

Map Names

Under the other options on this menu you'll find a list of all open maps. To switch between them, just click on the one you want. You can keep full-screen multiple map views and switch back and forth by clicking on the appropriate map name.

Note

Most StarEdit commands have keyboard shortcuts. Using these lessens dramatically the time it takes to create a map and campaign. Consult this book's appendix for a full list of keyboard shortcuts.

Help Menu

This menu offers general help for using StarEdit, and displays the licensing agreement and legal information regarding *Starcraft*, StarEdit, and any maps, scenarios, and campaigns you develop. It also lists contact information for Blizzard Entertainment (developers of *Starcraft* and StarEdit), as well as basic information about StarEdit itself.

Chapter 3 Building a Melee Map

This chapter will familiarize you with the StarEdit interface and teach you how to create a basic melee map. Once you build a simple map of your own, you'll be ready to delve deeper into the realm of map customization.

Tip

This chapter's instructions are explicit, but if you need to know more about a particular command or option, StarEdit's built-in Help function is excellent. Press F1 to access this menu.

The Groundwork

You'll be amazed at how easy it is to create a simple melee map. Because the computer handles so many factors automatically in melee games, you aren't required to do much work. Let's get started.

1. Select the Starcraft Campaign Editor (StarEdit) from your *Windows 95* (or *98*) Start menu (figure 3-1). The program will launch.

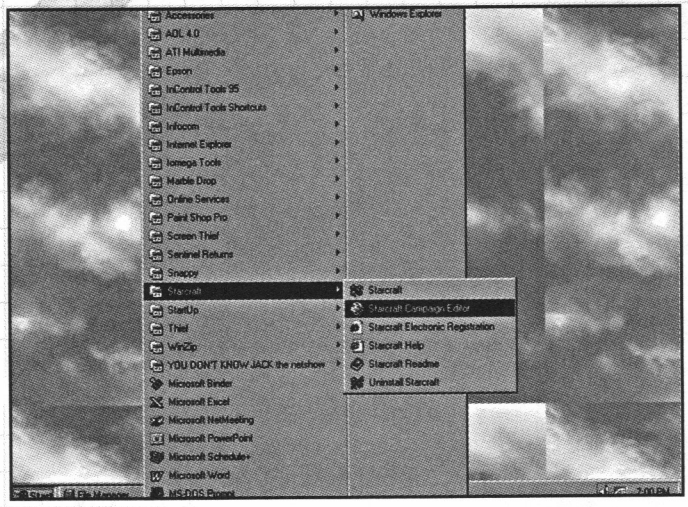


Fig. 3-1. Launch the Starcraft Campaign Editor to get started.

2. The program will open to a blank map named "untitled.scm". Examine the interface and pull-down menus. Slide the scroll bars to move the blank map. When you're done getting used to the interface, proceed to Step 3.

3. Under the File menu, select New. A dialog box like the one in figure 3-2 will appear. Under Tileset, select Space Platform. Leave Default Terrain as Platform. Finally, make sure the Square Map box is checked, and set Width to 64. When you've completed all this, press OK.

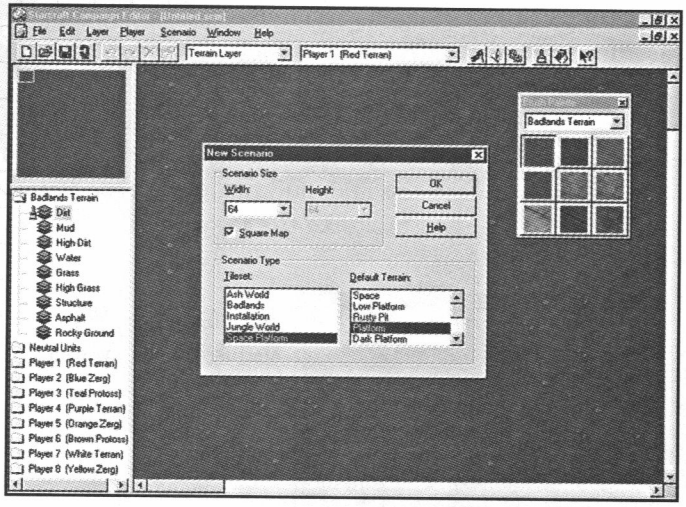


Fig. 3-2. This dialog box lets you set the size and general terrain type of your game world.

4. Now you've got a nice little 64-by-64 space platform—a great place for a small-scale space melee. Scroll to the map's upper-left corner. When you're there, click on the folder labeled Player 1 (Red Terran), on the screen's left sidebar. A number of new folders will appear. Click on Start Location, and then click on the icon labeled Start Location (figure 3-3).

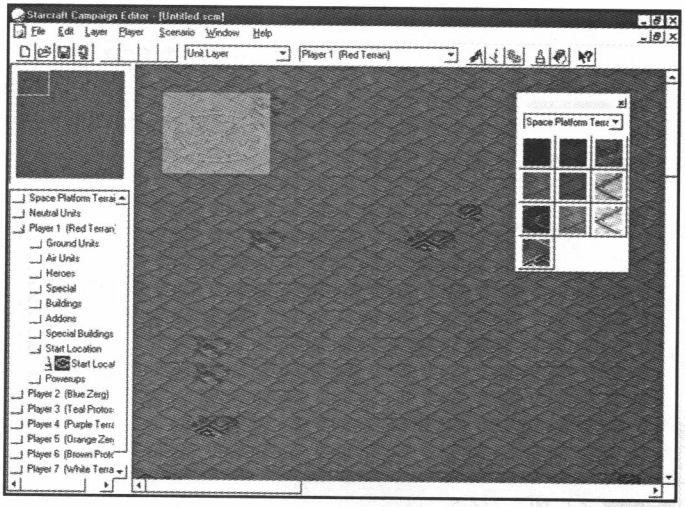


Fig. 3-3. Select a Start Location for Player 1 and place it on the map.

Tip

The New Scenario dialog box is fairly self-explanatory: Width and Height set the size of the map, and Tileset lets you choose its general look. (You can't mix and match tilesets.)

Default Terrain selects the terrain covering the map when you start editing. It's easiest if the Default Terrain is the type of terrain that will be most common on the completed map. For example, if you want to build a Badlands map mostly covered with water, select Water as the default terrain type. That way the water's there already, and much of your work is done.

5. Move the mouse cursor onto the map itself. You'll see a green highlight. Place the highlight somewhere near the extreme upper-left corner of the map and click. A red Start point appears for Player 1.
6. Now scroll down to the lower-right map corner. From the sidebar, select the Player 2 (Blue Zerg) folder. Select the Start Location subfolder, and the Start Location icon. Then move the cursor onto the map and place a Player 2 Start location near the lower-right corner. It will look like the Player 1 Start point, except it's blue (figure 3-4).

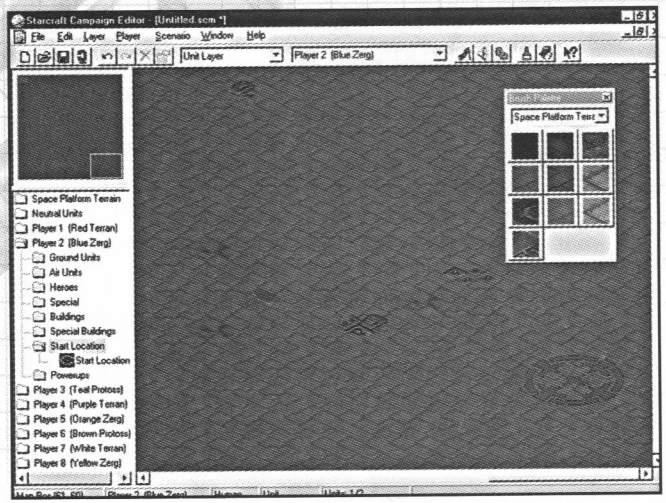


Fig. 3-4. Place the Player 2 Start location on the map.

7. Select the Neutral Units folder from the left sidebar, and then select the Resources sub-folder. Click on Mineral Field Type 1 and "paint" a small area near the Player 2 Start position with minerals (figure 3-5). You also can use Mineral Field types 2 and 3. They're functionally identical to Type 1; they only look different.

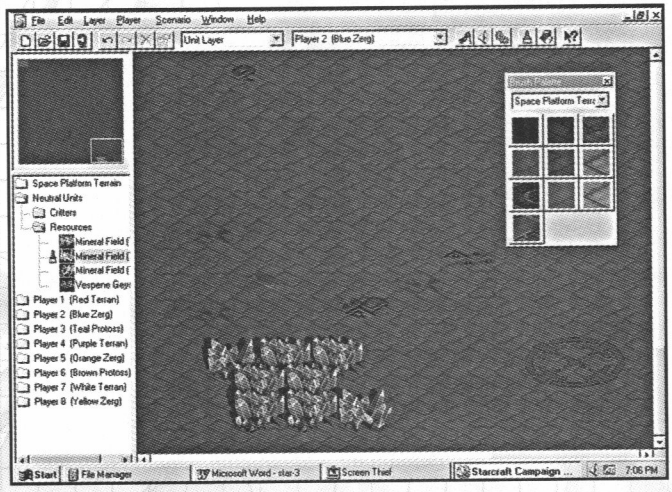


Fig. 3-5. Place a cluster of minerals near the Player 2 Start location.

8. Now select Vespene Geyser from the sidebar and drop a pair of Vespene Geysers near the Mineral Field (figure 3-6).

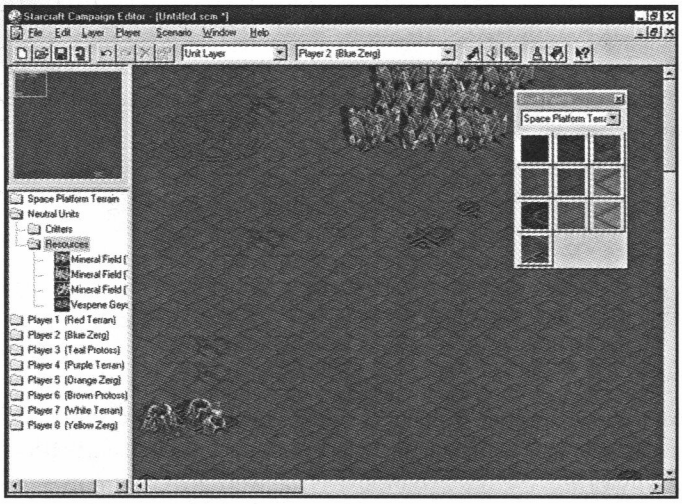


Fig. 3-6. Drop a Vespene Geyser near the minerals.

9. Scroll up to the Player 1 Start point and place some minerals and Vespene Geysers near it. Try to put roughly the same amount of resources there that you put near the Player 2 Start position.
10. From the File menu, select Save and give your map a file name, such as "test." Click Save to save the map.

Amazingly, your level is already playable as a melee map. To play it, start up *Starcraft* and select a single-player game. Enter your name, and then select Play Custom. Select test.scm from the scrolling list of missions, make sure Use Map Settings is enabled (figure 3-7), and then have at it!

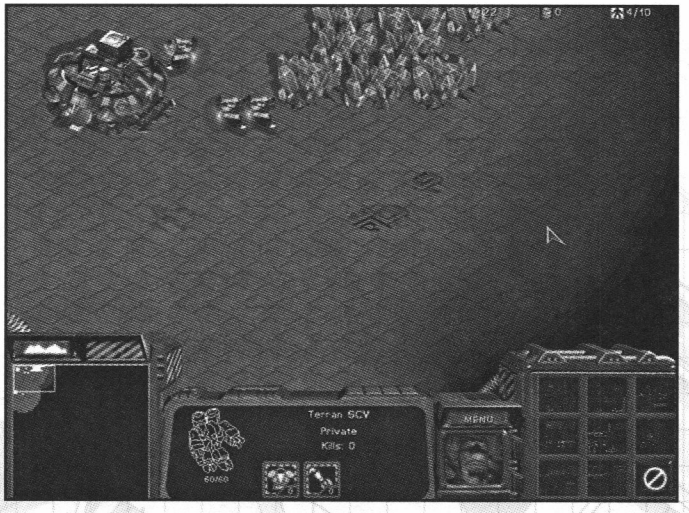


Fig. 3-7. Play your test map with the melee game type enabled.

Tip

If you paint too many minerals and want to erase them, you can get rid of them quite easily. First, select Unit from the Layer menu at the top of your screen to select the Unit layer. Then click and drag a box around the unwanted minerals. To get rid of them, either press Delete or right-click anywhere on the map and select Clear from the pop-up menu.

You can use the same process to delete any other unwanted units or resources. Select Terrain from the Layer pull-down menu to return to editing terrain.

Of course, you haven't made a particularly interesting map. It's small and perfectly flat, with no obstacles or other terrain features to complicate player strategy. There also are no resources, except those next to each player's base. This means players won't have to deal with the "resource grab" typical of most multiplayer games, when players seek new resources after depleting the most convenient supplies.

Fortunately, you're just getting started. Now that we've seen how easy it is to build a functional map, let's make this map a little more interesting.

Note

Note that the minimum requirements for a melee map are a pair of Start locations; you need no resources. However, without resources, the game is pointless. Each player starts with four worker units (SCVs, Drones, or Probes) that, in the absence of resources, can do nothing except attack the enemy's four worker units. Pretty dull.

Each starting location should have the same amount of resources, and the starting positions should be equally distant from the resources near them. You must ensure that each player has the same access to resources so as not to give one player an advantage over another. The exception to this is scenarios where you are *intentionally* giving the player a better or worse starting position.

Making it Better

Go back into StarEdit and load your map. At this point it's just a small, featureless slab of Space Platform. There's nothing interesting—nothing that forces players to approach this map any differently than they would any other. It's time to change that and give the map a little character.

Base Defenses

The first order of business is to add some defenses against extremely early strikes. Normally a small map like this one results in very quick games, but now you'll modify it to make both bases easier to defend.

1. Load the map and scroll to the upper-left corner, near the Player 1 Start location.
2. Select Low Platform from the Brush Palette floating in the upper-right corner of the screen.
3. Now you're going to use the Low Platform to build a sort of moat around the Player 1 Start point. Refer to figures 3-8 and 3-9 to see what the final product should look like, and then try to paint a line of Low Platform that mimics that shape.

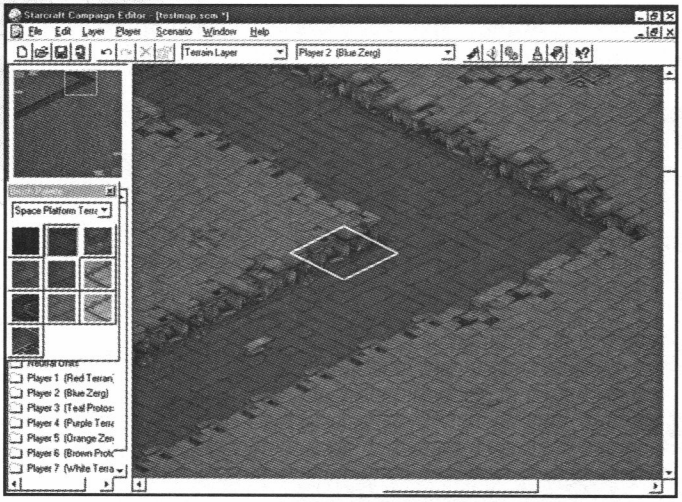


Fig. 3-8. Instead of creating wild shapes, build a moat from two straight lines of Low Platform.

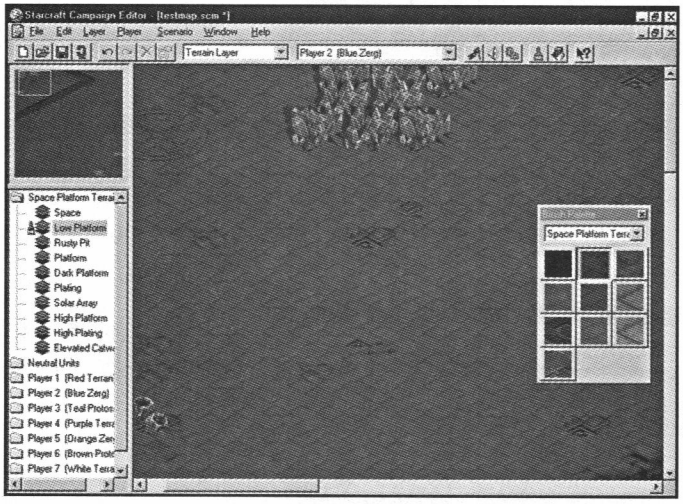


Fig. 3-9. This close-up of your Mini Map shows the shape your Low Platform moat should take.

The two moats don't need to be identical, but you must make sure that the Terran and Zerg players have enough room to build their core structures.

Making your Low Platform trench with straight lines rather than uneven ones makes the next few steps much easier. Now you have a trench or moat that separates the Player 1 Start position from the rest of the map. But you don't want to isolate Player 1 completely; you just want to make it difficult to attack him. It's time to add ramps.

4. From the Layer menu, select Doodad. The Doodad Palette will appear on the right side of your screen (figure 3-10).

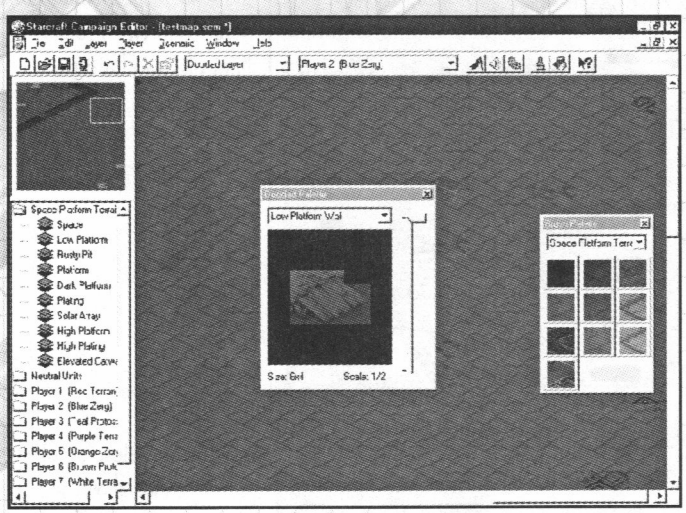


Fig. 3-10. The Doodad Palette lets you place various non-unit objects (called Doodads) on the map.

5. From the Doodad Palette, select Low Platform Wall. The Doodad Palette will show a ramp.

6. Click on the ramp in the middle of the Doodad Palette, and then move the cursor onto the Low Platform moat you just made. Place the ramp in the Low Platform (figure 3-11).

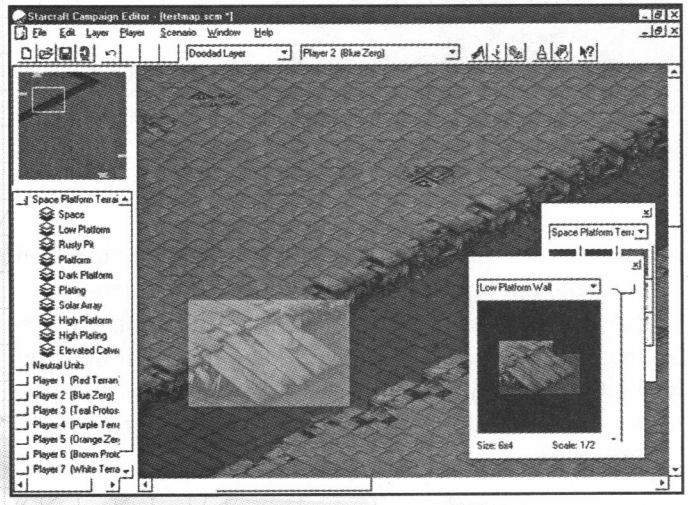


Fig. 3-11. Line up the ramp so the box around it is fully green. Click to place it.

Doodads

Doodads are non-unit objects that usually serve as simple obstacles or decorations. But a few, such as stairways and ramps, perform vital functions, such as allowing players' units to get from low terrain to high terrain, and vice versa.

These extra-functional Doodads, including stairways, ramps, bridges, doors, and traps, can be especially tricky to place. Usually they must sit on a certain terrain type: stairs and ramps, for example, must be stuck on a straight section of wall. Bridges are even trickier, and must be placed over a gap of perfect width.

If you find a particular Doodad tough to place, select it and hold it over the map to figure it out. Eventually you'll learn what sort of terrain you need to get it to work.

7. Now select the other ramp from the Doodad Palette by moving the sliding bar down. Place it in the Low Platform moat. You must place it at a different angle from the first ramp. You should also consider the size of the units that will be using the ramps and make sure that they have enough room to a maneuver. You want your Siege Tanks to be able to negotiate this terrain as well as your Marine does. Refer to figure 3-12 to see where to place it.

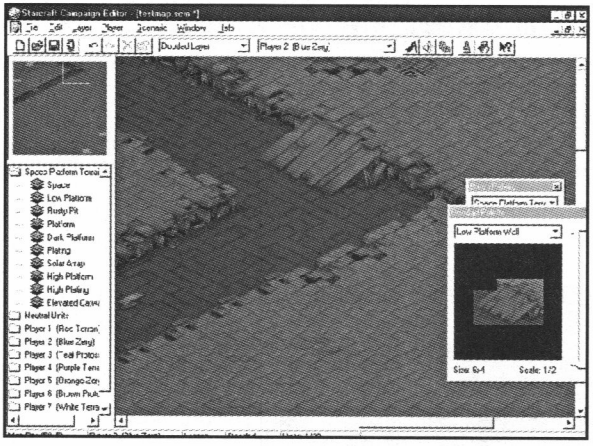


Fig. 3-12. Place the second ramp in the Low Platform moat.

8. Now repeat steps 1–7 for the area around the Player 2 Start location (figure 3-13).

Fig. 3-13. Build a more-or-less identical trench around the Player 2 Start location, complete with ramps.

Save the whole thing. You've created a trench around each base, with ramps leading up to each side. For land-based units to assault either base, they would have to descend a ramp, move a long distance through the trench, and then climb a ramp to the other side.

It's become more difficult to stage an attack on either base. Players can station defensive forces above the trenches to pelt oncoming enemies with weapons fire, and units positioned at the tops of ramps can block intruders physically.

With this single change, you've turned the map from a wide-open slugfest into a defensive struggle.

The Central Prize

Now that both bases are substantially harder to attack, you'll complicate matters by adding a central resource deposit. This deposit will feature both minerals and Vespene, and will be somewhat more prolific than the resources placed near the two Start points. It also will give the players a reason to leave their well-fortified bases.

1. Scroll to the center of the map and build a small, square area of High Plating (figure 3-14). Make sure it lies at dead center, or slightly above center, of the map.

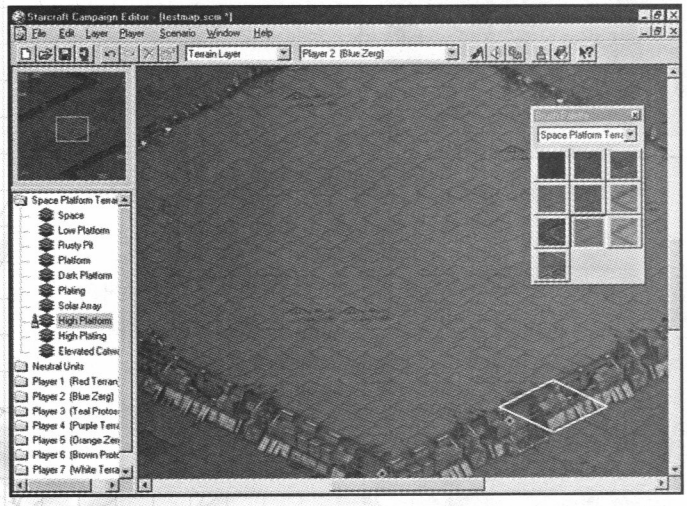


Fig. 3-14. Build a raised area in the middle of the map.

2. Switch to the Doodad layer and select Platform Wall from the Doodad Palette.

3. Scroll down to the fifth Doodad on the Platform Wall list—a staircase. Select the staircase and add it to the square area of High Platform you just built (figure 3-15).

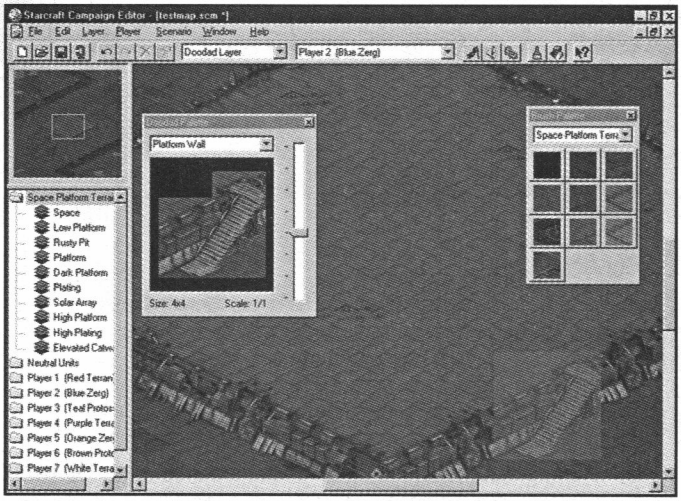


Fig. 3-15. Add staircases to make the High Platform accessible by ground units.

4. Scroll down to the sixth Doodad on the list, a staircase that attaches at a different angle. Select it and add it to the other edge of the High Platform.
5. Now add a cluster of several mineral deposits and a single Vespene Geyser to the top of the High Platform (figure 3-16).

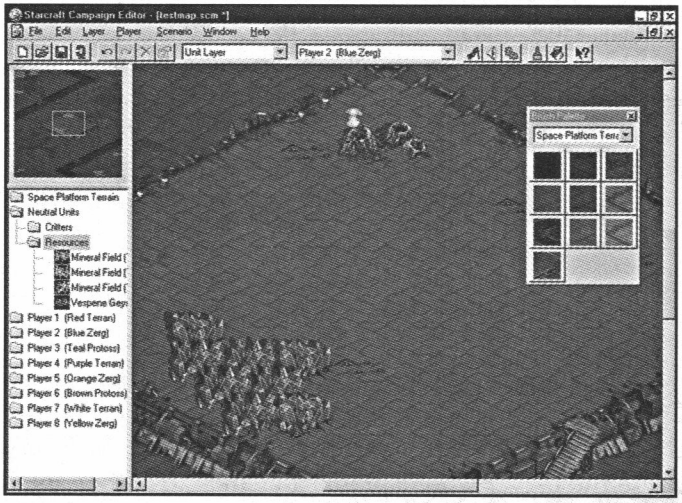


Fig. 3-16. Add some valuable resources to the top of the platform.

6. Right-click on the map to stop placing resources. Then click on one of the Mineral Fields you just placed on the High Platform. A green circle around it shows it's selected. Right-click on the selected minerals and select Properties from the pop-up menu.

7. The Mineral Field Properties dialog box appears. Select Resources, and change it from 1,500 to 3,000 (figure 3-17.tif).

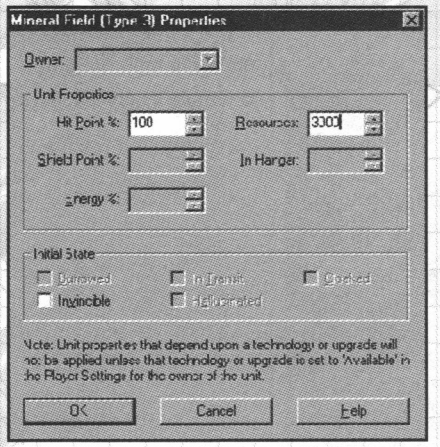


Fig. 3-17. Adjust the Mineral Field's Resource value so it yields more minerals when mined.

8. Now select the other Mineral Fields on the High Platform and change their Resource values to 3,000, as well. Also, select the lone Vespene Geyser and change its Resource value from 5,000 to 10,000.

You've just added a large, valuable deposit of minerals and Vespene to the map. Because these resources lie squarely in the center, they're up for grabs—either player can take them. By placing resources there, you give the center of the map tactical significance. Now this area is worth fighting for.

You could have added extra resources simply by dropping extra Mineral Fields and Vespene Geysers on the High Platform, but instead you dropped only a few Mineral Fields and a single Vespene Geyser, and then adjusted those units' values upward so they'd yield more resources. Either method is valid. The latter allows lots of resources without overcrowding the High Platform.

Outlying Fields

Now you can scatter a few extra resources around the map. These small Mineral Fields should be big enough to warrant building at least a Command Center, Nexus, or Hatchery and defensive structures to protect it. These resource sites will provide the player who explores a chance to gather some additional material by investigating the entire map.

There's nothing fancy involved. Just select a few spots for several Mineral Fields and perhaps a Vespen Geyser (figure 3-18).

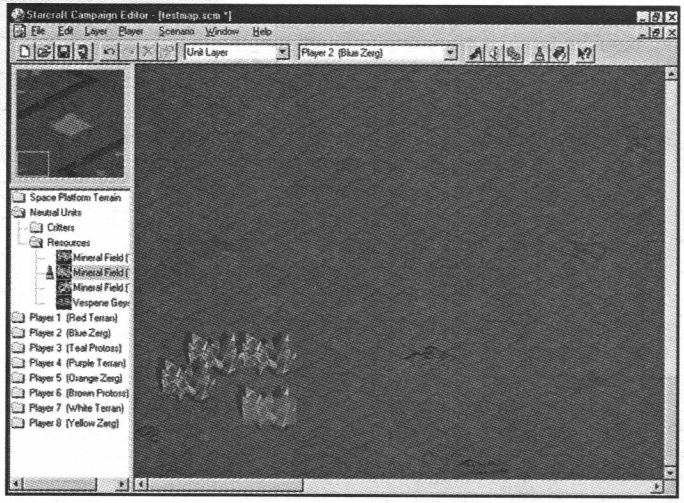


Fig. 3-18. Drop a few bonus Mineral Fields around the periphery of the map, just to complicate matters.

Giving it Character

Now your map is largely in place. The key features are there, but it could use a little detail. We'll use different terrain types and Doodads to help make the map look less uniform.

1. In the Terrain layer, select Dark Platform from the Brush Palette.
2. Paint areas of Dark Platform onto spots that used to be just Platform (figure 3-19). Dark Platform is identical to Platform functionally, but the color variation adds visual interest and gives the Platform a "worn" feel. It's all right to paint over units such as resources; they'll remain intact.

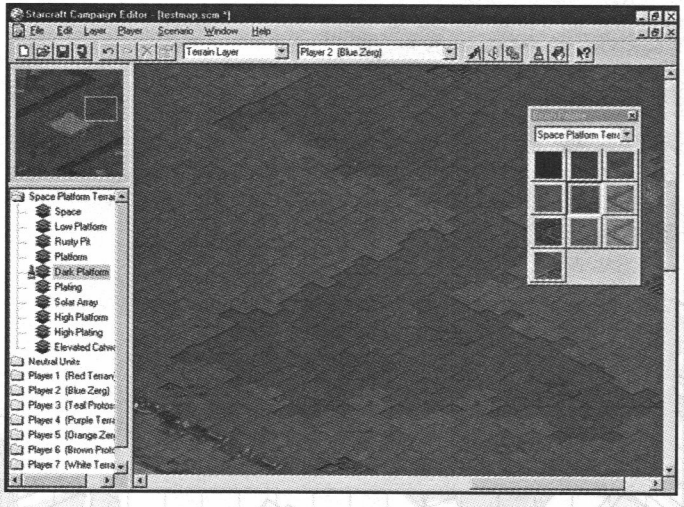


Fig. 3-19. Use Dark Platform to spice up the level.

3. Now select High Plating from the Brush Palette and paint the center of the High Platform in the middle of the map. Make it look like figure 3-20.

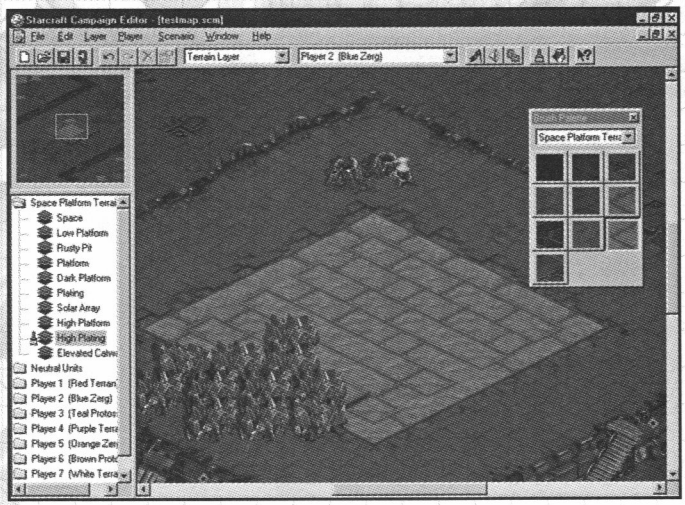


Fig. 3-20. Adding High Plating to the High Platform gives it a more finished look.

4. Go to the Doodad layer and select Platform from the Doodad Palette. Then select any interesting-looking Doodads you'd like to use and place them on the map (figure 3-21). Just don't go overboard!

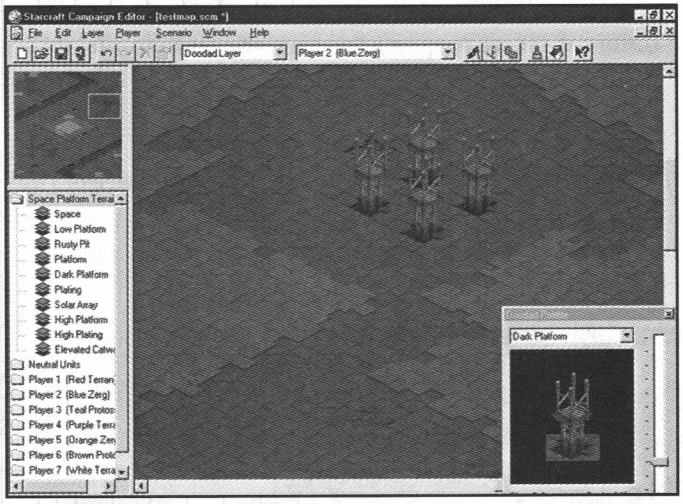


Fig. 3-21. Place those Doodads wherever you'd like.

5. If you *really* want to wreak havoc with the terrain, add a few small patches of Space or Rusty Pit (both available from the Brush Palette).

Continue playing with your map until it looks just the way you want it to.

Playtesting

At this point, your level looks the way you want it to, but it may not *play* the way you expect. If you really wanted to make this a good level, you will need to do some playtesting.

Playtesting involves playing the map several times, testing different strategies, and looking for flaws in the map. Here are just a few of the flaws you might find on this map:

- The Low Platform trenches guarding each base are too restrictive, and neither player can build a decent base without expanding.
- There are either too many resources in the central area or too few.
- The central resources are easier for one player to reach than the other, and so you must adjust their placement.
- Make sure that each player has equal resources, and that these resources are easily accessible. Ensure that no player has an advantage because of start location, unless this was intentionally part of the scenario design.

Unless you play a map extensively, you won't know what problems are associated with it.

Playtesting is hard work, but it's vital to ensure your final product is playable, fair, and—above all—fun.

Other Possibilities

You've learned the fundamentals of making a melee map. It's up to you to experiment. Here are some common variables that you can adjust to make each map play differently.

Map Size

The map you designed in this chapter was the smallest possible size. You can make much bigger maps, however, and significantly change the flow of the game.

Generally, small maps are all about speed and survival. You must get in gear and produce combat units early, because your enemy is never very far away. Often it's best to sacrifice long-term investments, such as high-level structures and units, for a horde of Terran Marines or Zerglings. Small maps usually make for short games.

Large maps, on the other hand, introduce new elements. It's still possible to win with a quick knockout punch of inexpensive units, but generally you must plan for the long haul. Large maps, more than small ones, give rise to full bases with lots of high-level structures and extremely potent units.

Exploration and expansion become more important on large maps. A major key to victory lies in scouting the terrain as fully and as quickly as possible, and then expanding to take over unclaimed resources and extend your sphere of influence.

Finally, large maps are far more suited to games with three or more players than small ones are. Packing four players onto the map we just created probably would be fun, but it would be over very quickly, as well: the game would take only a few minutes, because, with all those players on such a small area, players would bump into each other in no time.

Extra Players

It's easy to add extra players. Throw in a few extra Start locations, and they're good to go. Extra players complicate melee and free-for-all games immensely, adding to the fun.

The only complicated thing about adding extra players is ensuring that everyone gets a fair shot at winning. For example, in the map you just made, it would be unfair to put new Start locations in the northeast and southwest corners of the map, and not give those new players the same benefits you gave players 1 and 2. To keep the game fair, you'd need to give the new players a defensive trench and starting resources, as you did the first two players.

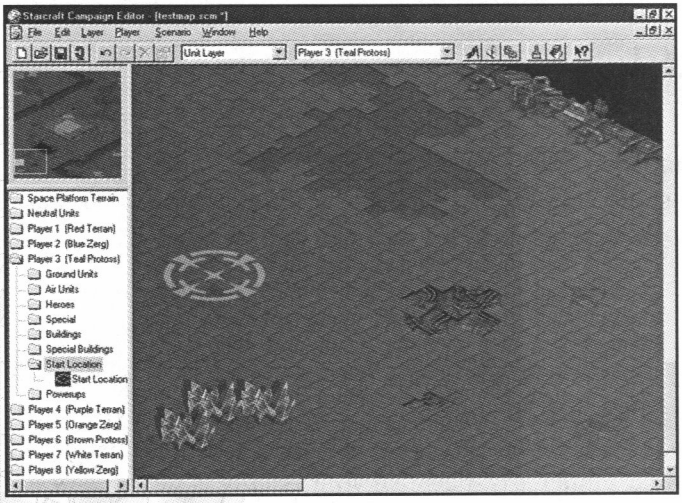


Fig. 3-22. If you plan to add a third or fourth player to this map, make sure they have some resources.

The more players you add, the more flexible the map will become. The map you just made is suitable only for one or two human players. But with two extra Start points and appropriate map modifications, you could make it suitable for games with up to four players.

Resource Distribution

Another way to add complexity to a map is to change its total resources and their distribution. Resource distribution can have a profound impact on the game.

There are only a few hard and fast rules here. First, before you start playing around with resources, remember: The computer AI can't handle situations where there are no resources close to its Start point. Always place resources near Start positions that might be owned by computer-controlled forces, just to make sure.

Also, always be sure to give each player an equal opportunity to grab resources. If one player doesn't have as many resources available as the others, he is at a huge disadvantage.

Typical maps feature a small amount of resources near each starting base and large amounts of resources scattered throughout the rest of the map. If the resources are clustered in one or two huge clumps, players naturally will fight over those clumps. If the resources are more scattered, players can go longer without conflict.

Experiment with different resource distributions and watch how they change the face of your mission.

Complicated Terrain

Terrain plays a big role in strategy. By adjusting terrain, you can change completely the way a game is played. Here are a few examples.

- Pits, gaps, water, and other impassable features force players to take set paths across the map, or to use flying units to bypass the terrain.
- High platforms can do much the same, but the higher terrain offers a tactical advantage if it can be reached.
- Doodads and other obstacles can force ground units around them and prevent base expansion.
- Creating Start locations or expansion locations completely cut off from the main part of the map forces players to develop advancements that allow for troop transport.

Asymmetrical Maps

Whether asymmetrical in terms of shape or in terms of resources, these maps offer interesting possibilities.

Shape Asymmetry

Maps built on an oblong or rectangular grid usually are the same as those built as square maps. But the feel of the two map types is very different. Long, thin maps tend to be head-to-head battles, with one army starting on one side and the other at the opposite end. Because of their shape, these maps don't work as well for multiple players; those starting on the nearer sides of the map are too close together.

Fig. 3-23. Many of the game's special scenarios are asymmetrical.

In scenario terms, a long, thin map works well to force a player from one side to the other, perhaps to retrieve or transport something from one place to the other.

Another possibility is terrain asymmetry. Giving the human player or a computer opponent a base that's difficult to access or leave either makes the mission more difficult or significantly easier.

Resource Asymmetry

Asymmetrical resources are a bad idea in most melee games, unless one player is significantly better than the other. Most of the time, you want to keep things equal between human players.

For single-player scenarios and campaigns, however, there's a real reason to give the human player more or less than the computer-controlled opponents. Giving the player a surplus, or placing resource caches near the human player's Start location makes the scenario easier: the player can build more units and reach the upgrade and special abilities more easily. Fewer resources means more choices, because not all the upgrades can always be developed.

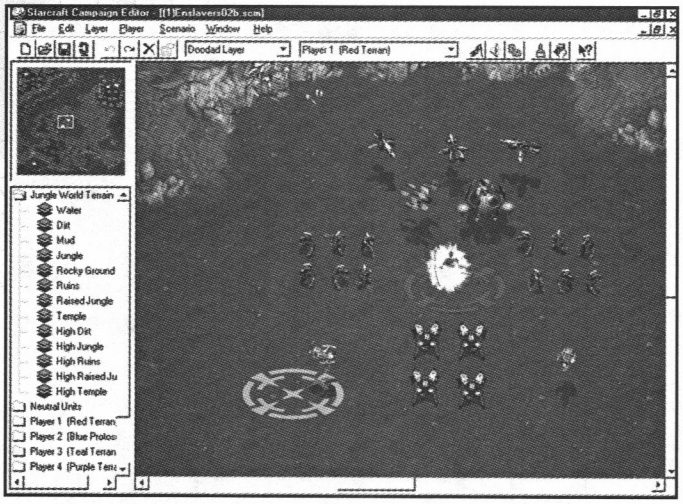


Fig. 3-24. You can give certain enemy groups, such as this Protoss force, few or no resources to make them more vulnerable to other players.

Or you can give the computer opponent very few resources, but start it with numerous units and structures already in place. Although the human player starts in an inferior position, it's more difficult for the computer opponent to replace losses.

Playing with the resources available to the human player and the computer opponents is an easy way to control scenario difficulty.

Chapter 4
Editing Your Scenario

In this chapter, you'll learn to customize your scenario up to the point where triggers become a factor. Many of these changes will be included in your scenario when you choose "Use Map Settings." Although triggers are the real heart of the StarEdit system (the following three chapters cover them extensively), the options this chapter explores give you some ways to truly customize your scenario.

Using Your Melee Map

Working with the melee map you created in Chapter 3, you'll develop a simple scenario that pits a Terran force against a group of enemy Zerg. Open your map and save it under a different name to use with this chapter.

The Terrans have developed a new Marine unit far more powerful than their normal Marine unit. Because you want the player to use this unit often, you'll eliminate some options that normally might be allowed, but you'll beef up this unit's power dramatically, as well.

So far, your map has no units or structures. Place a Terran Command Center in the upper-left corner, with four SCVs. At the lower right, place a Zerg Hatchery and four Drones. Save the scenario.

Options with Player Settings

Click on the Settings tab under the Player menu to bring up the Settings box. There are four submenus within this main area, all useful for modifying your scenario significantly.

Properties Menu

When you click on the Settings tab, this default box appears. Here you can alter the species of the armies in your scenario, as well as the way the player controls them when playing the scenario.

Species Options

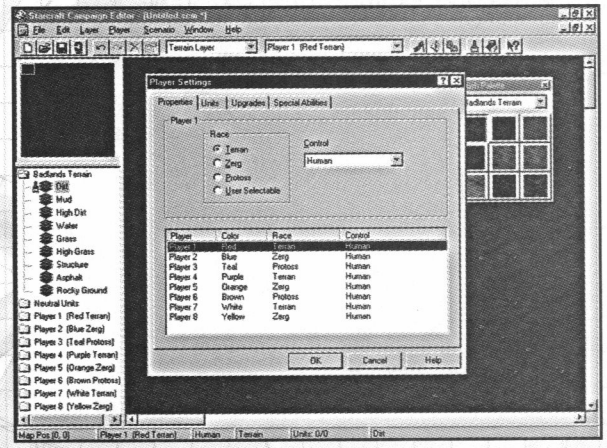


Fig. 4-1. Determining which species are active is one of the most important decisions you make as you develop your scenario.

Although only three species come with *Starcraft* and StarEdit, you may choose from four species for the armies in your scenario. The first three choices—Terran, Zerg, and Protoss—are self-explanatory. The fourth allows the *player* to pick a species to use in the scenario.

The User Selectable option is best used in melee maps that have no goal beyond destroying the opposing force. This way, people playing on your map can choose the army they're most comfortable with. Many missions you'll create will have specific goals for your players. Many of these goals will require that the player be of a specific species.

Tip

You can't make a computer-controlled force or army user-selectable. This option is reserved solely for human-controlled armies.

Don't select species on a whim. How you choose the species in your scenario will determine much of the way you develop the rest of your mission, particularly if you're going to create something beyond a simple melee map. Pitting a Terran player against a Zerg force will differ greatly in terms of potential story and strategy than Terran against Terran. Before you choose which armies to use, try to have a basic idea of what you'd like your scenario to entail.

Go back to the map you made in the last chapter. You've included Start locations for the red army (default Terran) and the blue army (default Zerg). If you'd rather make this battle Zerg against Protoss, you need only change the Player 1 species to Protoss. To use different colors, add a Start location for one of the other colors and change the species accordingly.

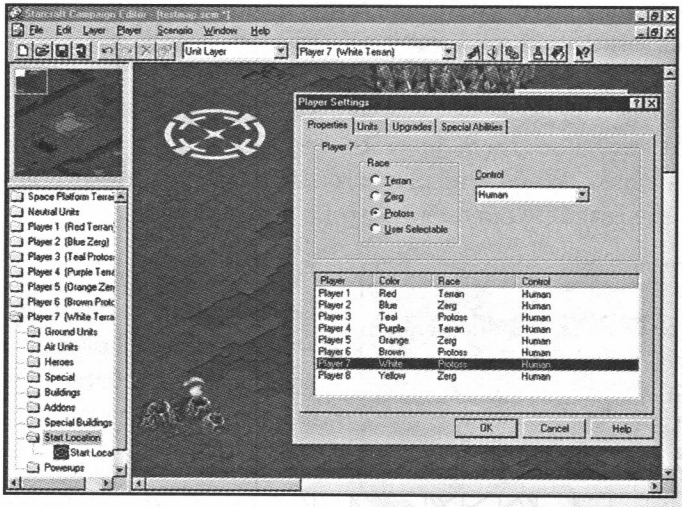


Fig. 4-2. A Start location has been added for the white army, and the force changed to Protoss. If you play this mission now, it will pit blue Zerg forces against white Protoss forces. Play around with this, but don't save over your melee map.

Control Options

You also may choose the control method—computer, human, rescuable, and neutral forces.

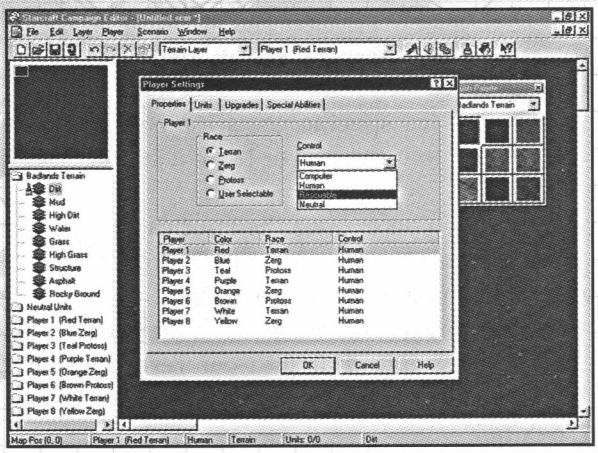


Fig. 4-3. Each of the four kinds of control functions differ in a scenario.

Human and computer-controlled forces are precisely what they sound like. Computer-controlled forces always will be controlled by a computer opponent. Human-controlled forces always will be controlled by a human player, unless there aren't enough human players to fill every slot. For instance, many scenarios will be available for one to four players, or two to five players. Each human player will control an army; if there are more armies than players, the forces left over simply won't be included.

Switch the blue Zerg forces to computer control and save your scenario.

Rescuable forces operate differently. These units neither move nor attack; computer-controlled forces ignore them. Instead, these units sit in place until a human player approaches. When the player contacts them, these units become active, under the rescuing player's control.

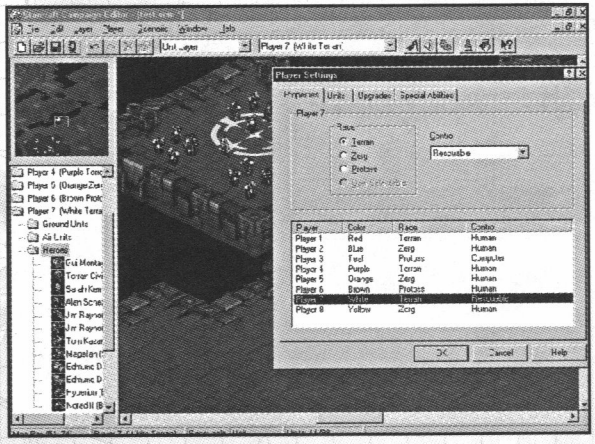


Fig. 4-4. These Terran units are designated rescuable. The first human player to reach them will gain the services of nine Marines, three Firebats, and Kerrigan—a worthwhile discovery!

It's also possible to rescue structures. Any rescuable structure a human player contacts becomes the property of that player. In fact, rescuing a colony center (Terran Command Center, Zerg Hatchery, or Protoss Gateway) gives the rescuing player control of all structures and units connected to that building.

Rescuable units often become the centerpiece for an entire scenario. Reaching a rescuable group can be goal of a mission, or can give a player the firepower necessary to succeed.

A neutral force acts on its own, without attacking other players or forces. If a neutral force is attacked, however, it identifies the attacking player as an enemy, and becomes a hostile force. For the rest of the scenario, this force will defend itself against that player.

Units Menu

This menu allows you to control the units available to a given army, increasing or decreasing scenario difficulty significantly. To give a human player an advantage, for example, you can eliminate a computer enemy's ability to build some powerful units. Or to increase the scenario's difficulty, you can deny a human player the ability to create certain units.

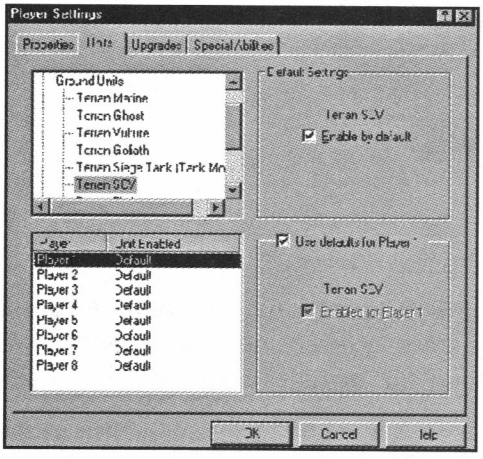


Fig. 4-5. From this menu, you control the units a given player, or an entire scenario, can generate.

All units for each army are enabled by default. That is, if you make no changes here, player(s) can develop every unit available to their force.

To change a given unit's availability across the scenario, select it from the menu at the upper left. Clicking off the check box under Default Settings disables that unit across the board.

To limit just one army, select the player from the menu at the lower left and click off the Use Defaults box. This allows you to disable units for the selected player, whether computer-controlled, human, neutral, or rescuable.

Tip

Disabling a unit type across the board allows you to create "low-tech" missions, where the highest level of the research tree is unreachable. To give a weaker player an advantage against other human players, or to adjust your scenario's difficulty, you can enable or disable units for a given player.

For this scenario, you'll remove a few goodies from the player, because you want more reliance on the new Marines you'll create. In the Terran Units menu, click on Ghost, and then click off the Use Defaults button. To disable Ghost units for the player, click off the Enabled for Player 1 button in the lower-right part of the menu. Player 1 now is unable to create Ghosts in this scenario. Do the same thing for Siege Tanks and Firebats. Now go to the Addons folder under Terran Units and disable the ability to create Nuclear Silos and Covert Ops.

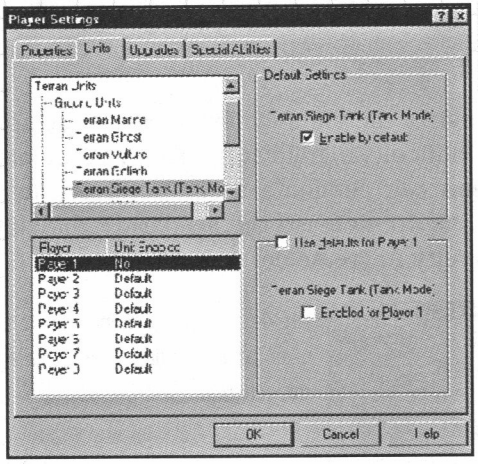


Fig. 4-6. You've disabled Siege Tanks for Player 1. The screen says "No" under Unit Enabled (lower left), meaning the player can't build this unit in this scenario.

Upgrades Menu

The Upgrades menu gives you control over your scenario's upgrades. From here, you control upgrade availability, starting level, and how many times it can be researched. As in the Units menu, all upgrades are enabled to maximum levels by default.

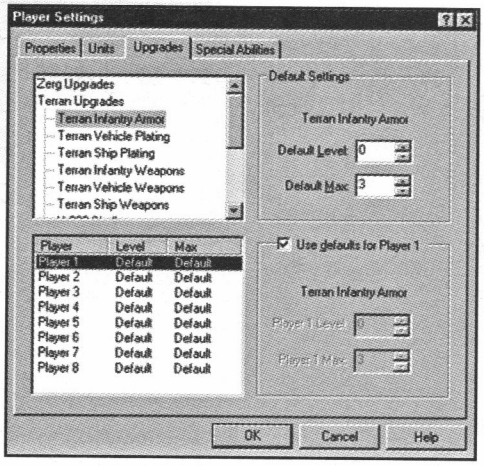


Fig. 4-7. Determine upgrade availability, starting level of technology, and each player's ability to improve with this menu.

This menu offers a few other options, but it functions essentially the same as the Units menu. To disable a particular upgrade across the board, highlight it on the upper-left menu. Now reduce the Default Max slider to zero to turn off the upgrade throughout your scenario. If you don't want players to be able to research a particular technology fully, use this function to reduce the maximum upgrade level. To start your players with the upgrade researched or partially researched already, adjust the Default Starting Level tab.

To control these options for a particular player, select that player and use the Default Max and Default Starting Level tabs in the lower right. To affect these levels for a particular player, first click off the Use Defaults for Player box.

Because you want the player to rely on those Marines, you'll disable a few upgrades. Remove the ability to research Apollo and Colossus reactors to prevent the player from increasing Wraith and Battlecruiser energy capacity. Because the player can't develop Ghost units, remove the ability to create ocular implants and Moebius Reactors (reduce Max Upgrade level to zero for Player 1).

Now select Terran Vehicle Plating and reduce Player 1's maximum upgrade level to 2. Repeat with Terran Ship Plating, Terran Vehicle Weapons, and Terran Ship Weapons. Now the player's vehicles and ships can get only two levels of upgrade.

Finally, you'll add power to those Marines: With Ghosts and Firebats disabled, the Marines become the only infantry units available to this player. Click on Terran Infantry Weapons and increase Player 1's starting level to 1. Do the same with Terran Infantry Armor. Now each Marine unit will be created with one upgrade level of armor and weaponry, and the player need pay for only two upgrades.

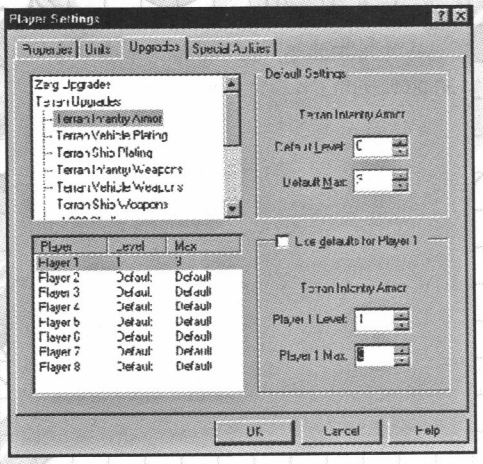


Fig. 4-8. The Marines are now more powerful: they start the mission with one level of upgrade in armor and weaponry.

Note

You can't increase a technology's maximum level beyond the default max. Terran Infantry Armor, for instance, never can be increased beyond three levels of upgrade.

Special Abilities Menu

This menu controls the units' special abilities. You have three choices for each special ability—disabling, enabling, or automatically researching. Unlike upgrades, special abilities have only one level of research: either you have the special ability or you don't. And, as with both Upgrades and Units menus, you can modify across the board or for a single particular player or group of players. All special abilities are defaulted as researchable, and none are researched by default at the beginning of a mission.

For the most part, this menu works like the previous two. Select the special ability you wish to adjust. To change it across the board, click on the appropriate box in the upper-right corner. Select Disabled by Default to prevent players from researching or using that special ability. Enabled by Default, the normal setting, allows players to research it. Researched by Default means any player using a species with that special ability will start the scenario with that special ability ready to use.

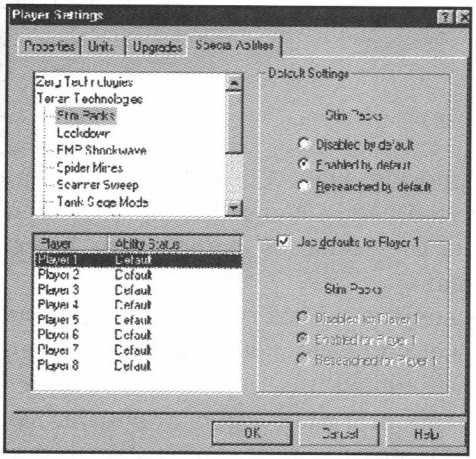


Fig. 4-9. Use this menu to affect the way players can use and research special abilities.

To modify these special abilities for a specific player, select the player from the menu at the lower left. Turn off the Use Default Settings for that player to change these settings. Then use Disabled, Enabled, and Researched by Default tabs at the lower right to affect settings for each player you wish to affect.

For your scenario, you'll disable a few special abilities. First, get rid of Lockdown, Siege Tank Mode, and Personal Cloaking—all special abilities for units excluded from this scenario. Then select Stim Packs and make them researched by default. Now the player needn't research this option, and the Marines he or she creates can use Stim Packs automatically.

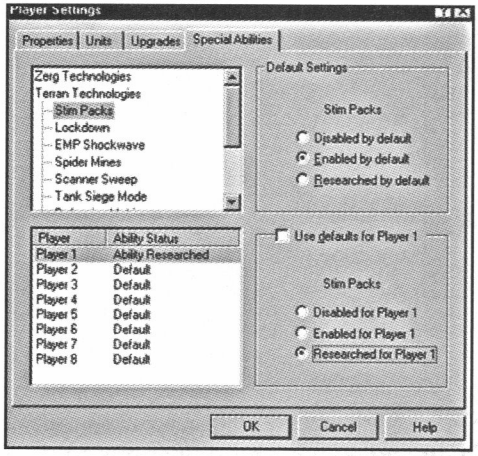


Fig. 4-10. Now Stim Packs have been researched for Player 1, and are turned on automatically in this scenario.

Note

Although they seem the same on the surface, there's a real difference between upgrades and special abilities. Upgrades physically improve your units in terms of weaponry, armor, energy storage capacity, and speed. Special abilities are a specific unit's powers—for example, the Zerg's Burrowing, the Terran Ghost's Lockdown, the Protoss Templar's Psionic Swarm.

Scenario Options

Forces Options

Now you'll divide your forces into opposing sides. Currently, you have a human player, playing as Terran forces (red), and a computer opponent, Zerg (blue). Click on the Forces option under the Scenarios menu (or the Forces button in the toolbar) to reveal this menu.

Notice that all sides start on the same force. Click on and drag the blue Zerg forces to the box under Force 2. Let go of the mouse button; the Zerg forces now appear under this heading. By doing this, you've made Player 1 and the Zerg armies enemies. Click off Random Start Location for both armies, because we want Player 1 to start in the upper-left corner and the Zerg to start at the lower right.

You're creating a scenario with only two armies, so click OK and move on, if you'd like. To clean up the forces a little, you can drag players 3–8 to Force 3 or Force 4, but this isn't necessary.

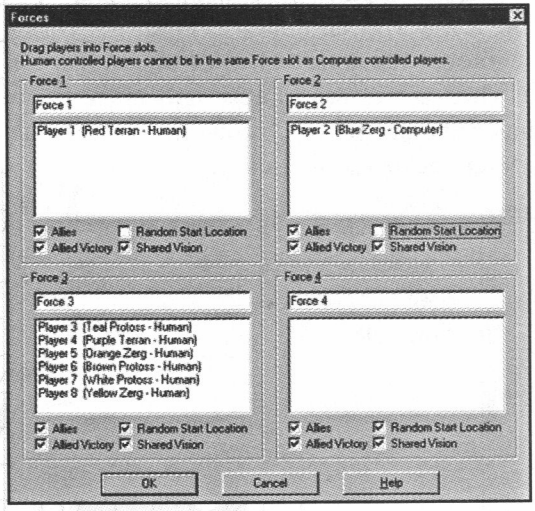


Fig. 4-11. Your sides are chosen, and you're ready to go.

Playing with Settings

Now comes the area where you can make some serious modifications to your *Starcraft* scenario. Where options in the Player Settings menu affect the availability of units, upgrades, and special abilities, the options we explore here control the *power* of those units, upgrades, and abilities. Again, you'll concentrate on the scenario you've been working on in this chapter.

Unit and Hero Settings

Here you'll adjust actual unit values. Chapter 2 covered this briefly, but here you'll put those principles into practice.

To change the settings for a given unit, structure, or add-on structure, you first must click off the Use Default Settings box in the upper left. Now you can increase or decrease unit Hit Point capacity, armor, shields (for Protoss units), resource requirements and time, and physical strength.

For your scenario, take a look at the Terran Marine. You're going to beef up this unit's power: what quality upgrade would justify a player being unable to create Ghosts, Firebats, and Siege Tanks?

Start by boosting the Marine's Hit Points. In the menu at the upper left, select Terran Marine, and then deselect the Use Default Settings box. Increase the Marine's Hit Points to 60; give the unit 1 point of armor, while you're at it.

To make the Marines even more deadly, you can boost the damage they inflict. Increase Ground Weapon Damage from 6 to 9, and the Bonus from 1 to 2. Now each weapons upgrade will increase the Marines' damage by 2 points instead of 1, making the Marines far more potent both offensively and defensively.

But this increase in power should come at some price. The time it takes to produce a Marine is 24 seconds: boost it to 36. Take the mineral cost from 50 to 65, and require 10 units of gas.

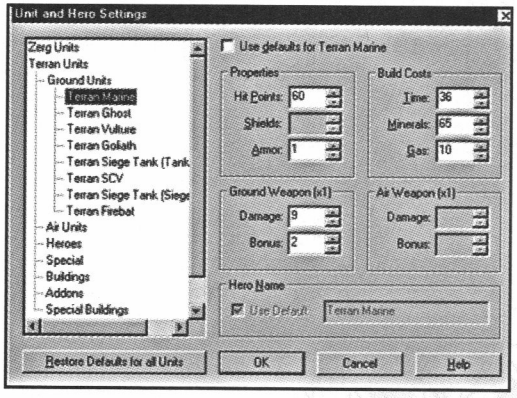


Fig. 4-12. The Terran Marine is now a far more powerful and interesting unit for this scenario.

Note

The values in the following menus are global changes; that is, they affect all units, upgrades, structures, and abilities of a given type across the entire scenario. For the one you're creating, this means additional Terran players will benefit from your upgraded Marines without the limitations you're imposing on Player 1—your human player. Obviously, if you wanted to add players to the scenario, you'd reconsider some of the changes you've made so far, or disable units, upgrades, and special abilities across the board.

Note

You added an automatic armor upgrade to these units in the foregoing section, so Marines will come with 1 point of armor and one upgrade, effectively giving them 2 points of armor.

You've taken the basic Terran Marine and souped him up dramatically. He has better armor and more Hit Points; plus, he's far more effective in combat—and the increased time and resource cost for producing this unit reflects it.

Note

Because you've disabled these units, you needn't touch the settings for Siege Tanks, Firebats, and Ghosts.

Upgrade Settings

From this menu, you'll adjust upgrade costs for the units in your scenario. Again, Chapter 2 covered this, and now you'll apply those principles to the scenario at hand.

You increased the infantry weapons bonus earlier, so it makes sense that this upgrade should become a little more costly. On the Upgrade Settings menu, select Terran Infantry Weapons from the menu on the left and deselect the Use Defaults box. Now you can adjust costs for this upgrade.

Because the upgrade bonus has been doubled, boost gas and mineral cost to 150 each. Increase the time it takes to produce the upgrade from 266 seconds to 330 seconds. Now the upgrade is more expensive *and* it takes more than a minute longer to research.

Additional upgrades also should be a little more expensive. Increase the time factor from 32 to 60, so each additional infantry weapon upgrade will take a minute longer to produce. Boost mineral and gas factors from 75 for each upgrade to 125, making each upgrade 125 units of each resource more expensive.

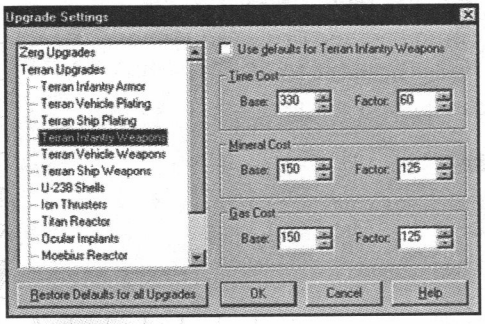


Fig. 4-13. Terran infantry weaponry is more expensive and time-consuming to upgrade, reflecting the increase in power you gave Marine weaponry.

ABILITY SETTINGS

Now you can adjust the costs of using and researching special abilities. Again, Chapter 2 covered this, and now you'll apply what you learned.

Other than disabling a few and giving the player Stim Packs by default, you haven't done a lot in terms of special abilities so far. Stim Packs is the only special ability that affects Terran Marines, so you'll need to play with a few values for other special abilities.

Remember, initially we wanted the player to rely far more on the Marines. Now you'll increase the cost for using other special abilities. Because you want to try and keep this a ground war, play with the costs of the Wraith's most powerful ability—Cloaking.

Select Cloaking Field from the menu on the left and deselect the Use Defaults box. You want to make this more expensive and difficult for the player of this scenario to use. The development time is fine, so increase the cost. Boost both mineral and gas cost for research to 200, making it 33 percent more expensive to research.

You also want to make it more difficult to use. Normally, the cost of initiating Cloaking is 25 units of energy. Increase that to 50: now a Wraith must have 50 units of energy simply to use its Cloaking ability, and it will require a huge percentage of the Wraith's energy.

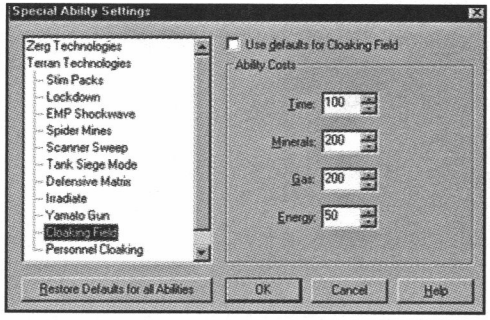


Fig. 4-14. Now the Wraith's Cloaking ability is more expensive to develop and more costly to use, limiting its effectiveness.

A Quick Playtest

Your scenario isn't ready to play yet, but it's worth checking out, anyway. If you investigate the map's lower-right corner, you'll notice the Zerg forces aren't doing anything: you haven't given them an artificial intelligence yet. So don't investigate that part of the map.

Instead, play around with your base, developing buildings and creating units. The changes you made from this chapter have become part of your scenario. The Marines are more powerful and more expensive, and you can't build Firebats, Ghosts, or Siege Tanks.

When you're satisfied that your changes "took," quit playtesting and move on to Chapter 5, where you'll add mission briefings and link missions to make campaigns.

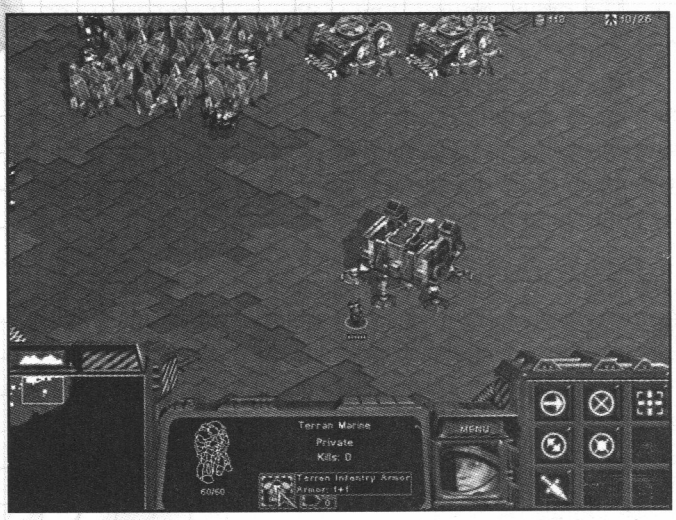


Fig. 4-15. This freshly created Terran Marine has 60 hit points, 2 points of armor, and the ability to use Stim Packs as soon as he walks out of the Barracks.

Chapter 5 Creating Mission Briefings and Campaigns

Creating Mission Briefings

If you played around with your scenario at the end of Chapter 4, you noticed that not much happened when you first started it up. There were no objectives, no mission briefing to start you.

There also will be times you'll want to create more than one mission based on a theme, to be played in a certain order—that is, a campaign.

Follow along, then, and this chapter will give you the skills you need both to create briefings and to link your missions.

Determining the Goal of Your Scenario

Before you can create a mission briefing, you must determine what you want your mission to be about. Your goals are limited only by your imagination (and the *Starcraft* engine: try as you might, you'll never turn it into a first-person action game or a flight simulator). Chapter 7 covers some of the more esoteric possibilities. For learning purposes, in this chapter you'll work with standard types of *Starcraft* missions.

In the last chapter, you created an upgraded Terran Marine to see how it would fare against Zerg forces. Your goal for the mission naturally will center around this. It could be as simple as having the player field test the new Marine specifications before changes are implemented throughout the Terran army. Or perhaps the Terrans have developed these more powerful Marines to deal with a particularly well-established and tough-to-destroy Zerg encampment.

Either possibility will work for the new Marines you've developed, as could several others. For now, you'll use the second as your mission's main objective. At the moment, however, you don't have a particularly large or powerful Zerg encampment. So you'll need to make several additions to your map to beef up Zerg forces and prepare them for an assault by your human player. Referring to Chapter 3, place structures and units for the Zerg forces. To keep Player 1 from being overwhelmed, put down some Terran structures and units, as well.

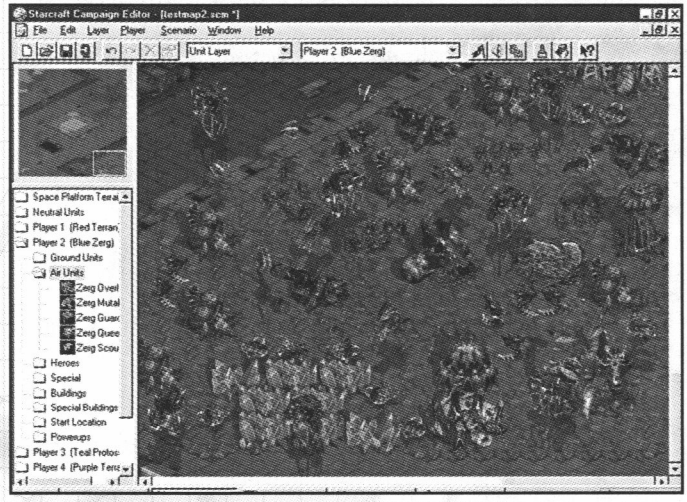


Fig. 5-1. Beef up Zerg forces to include a Hydralisk Den, Spawning Pool, Evolution Chamber, Hive, Spire, Extractor, and several Spore Colonies and Sunken Colonies. Add 24 Zerglings, eight Hydralisks, two Queens, eight Drones, four Scourges, six Mutalisks, and eight Overlords, as well.

Fig. 5-2. For the Terran player, you'll add a Refinery, Barracks, Engineering Bay, five Supply Depots, and a pair of Bunkers. You'll also give the Terrans a dozen Marines, a half-dozen SCVs, two Vultures, and a Dropship.

Creating the Mission Briefing

With your starting positions set up, you can create the mission briefing Player 1 will see when the mission starts. This is your first use of Trigger functions, and although it can get a little complicated, with practice the system becomes easy to use. (Remember, StarEdit's excellent Help menu is always available.)

How the Mission Briefings Work

Go to the Scenarios pull-down menu and select Mission Briefings. The large blank box you get may throw you, but don't worry: you'll fill it up in a moment.

Click on the New button to get a second pop-up box. Here you'll determine the player or players this mission briefing will affect. You're working on a single-player scenario, so click on Player 1 to tell StarEdit to show Player 1 this briefing at the start of the mission. Then click on the Next button.

Tip

When placing the building used to extract Vespene gas, you first must delete the geyser, and then place the building. You can't cover up a unit (here, the Vespene Geyser is a neutral unit) with another unit or structure.

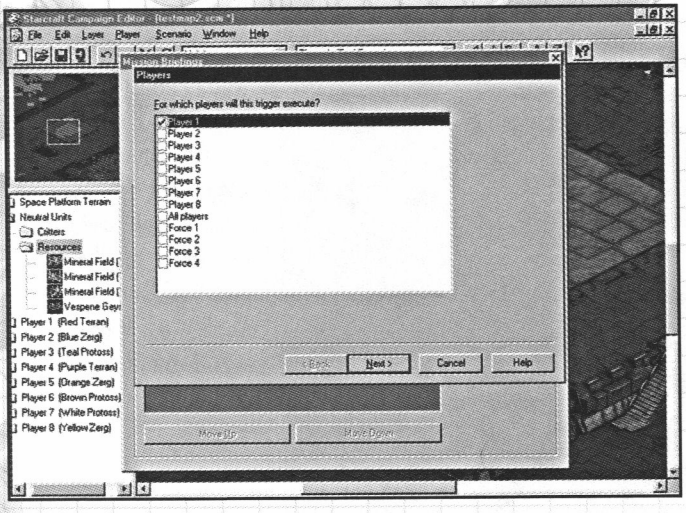


Fig. 5-3. Player 1 will receive this briefing every time this mission launches.

When you create scenarios with varying objectives for other armies, you can create different briefings for each player. You also can create briefings that play to all human players of a given force (refer to Chapter 4 for details on creating forces), or to all players, period.

You'll display this briefing to Player 1. Because this is your only human player, you could have selected All Players just as easily. And because your human player is in Force 1, created in the previous chapter, clicking Force 1 would have been appropriate, as well.

Now you have a blank menu labeled "Actions." To add an action to the briefing, hit the New button to view a list of possibilities. When you're satisfied with a particular action, click OK in the New Briefing Action menu to return to the Actions dialogue box. You can add as many actions to the briefing as you wish, hitting the New button for each. When you're satisfied with your briefing, hit the Finish button in the Actions dialogue box.

All this becomes less confusing once you start putting it into practice. But before you start creating your briefing, take time here to acquire a more thorough understanding of how triggers work.

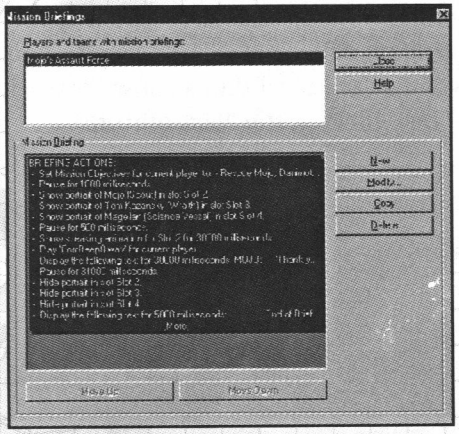


Fig. 5-4. A complete mission briefing consists of a series of linked actions. Before you can create a briefing, you must understand each action's potential.

Mission Objectives

The most basic briefing action, and perhaps easiest to understand, is Mission Objectives. This action places a text summary of actions a player must take to win the scenario in the briefing's lower-left corner.

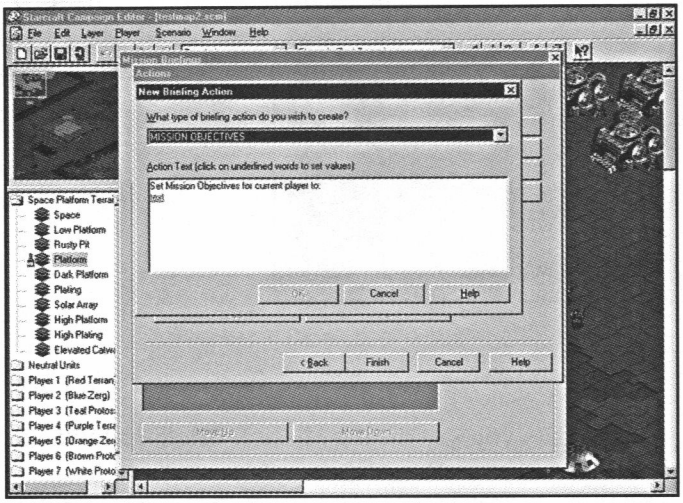


Fig. 5-5. The Mission Objectives Dialogue Box

To create Mission Objectives, click New in the Actions menu. In the New Briefing Action menu that appears, click on the pull-down menu and select Mission Objectives. This brings up a new dialogue box containing the text, "Set Mission Objectives for current player to:" with a bit of text highlighted blue. Click on the blue text. This generates another dialogue box where you may type in your mission objectives. For example, the scenario you've been working with might have objectives such as these:

- Test new Marine specifications.
- Find and exploit resources on central platform.
- Destroy all Zerg forces.

Click OK to return to the Actions menu. You're ready to create your next Mission Briefing action.

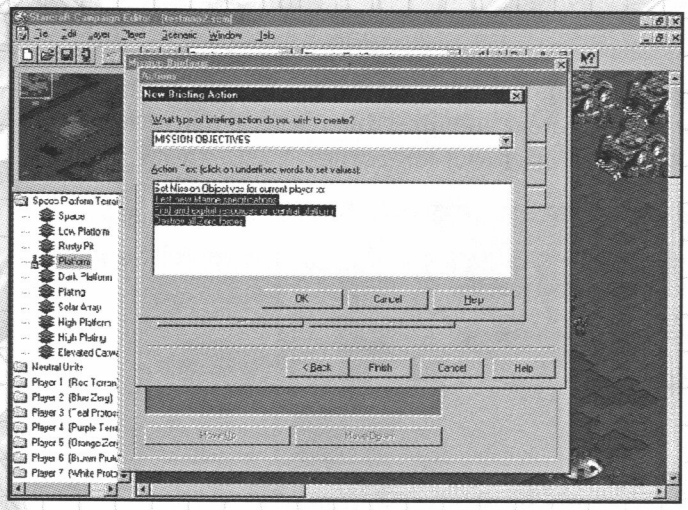


Fig. 5-6. You've set the Mission Objectives. When the scenario is played, they'll appear in the lower-left corner.

Showing and Hiding Portraits

You can control what the portraits you see during Mission Briefings look like, how long a player views them, and what they do.

From the pull-down menu on the New Briefing Action dialogue box, select Show Portrait. You'll be asked for a slot number and the portrait you wish to show. Select slot number first.

You may show the portrait in one of four slots, numbered 1–4. In the actual briefing, Slot 1 lies on the far left side of the screen and Slot 4 lies on the far right. Determine where you want the portrait to appear and select the appropriate box. For now, keep the default Slot 1.

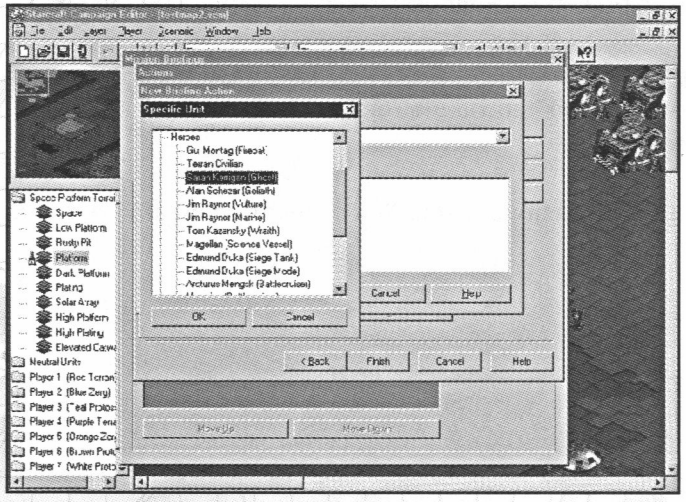


Fig. 5-7. Now Sarah Kerrigan's portrait will appear in the left window of the mission briefing.

Next, click on Unit to view a scrolling list at the left side of the resulting pop-up menu. You can place virtually any *Starcraft* image in the portrait slot you selected—buildings, units, enemies, heroes, resources, and so on. Select Terran Units, then Heroes, then Sarah Kerrigan (Ghost), and hit OK.

If you ran this mission now, you'd view a picture of Sarah Kerrigan in the leftmost Briefing screen window.

So you have a picture of Sarah Kerrigan in Slot 1. What else can you do with that picture? You can make it talk. Return to the New Briefing Action dialogue box and select Display Speaking Portrait from the pull-down menu.

As with showing a portrait, you have two things to determine here. First, you must select the portrait you wish to animate. Click Slot Number to view your options—slots 1–4. For now, keep the default of Slot 1 because this is the only slot where you've shown a portrait.

You must determine the length of time you want the picture of Kerrigan to stay animated. Click on Number of Milliseconds. For now, choose 10,000 milliseconds and click OK.

Caution

You *must* show a portrait before you can display the speaking version. Before using this command, you must include an action to show the portrait in the first place.

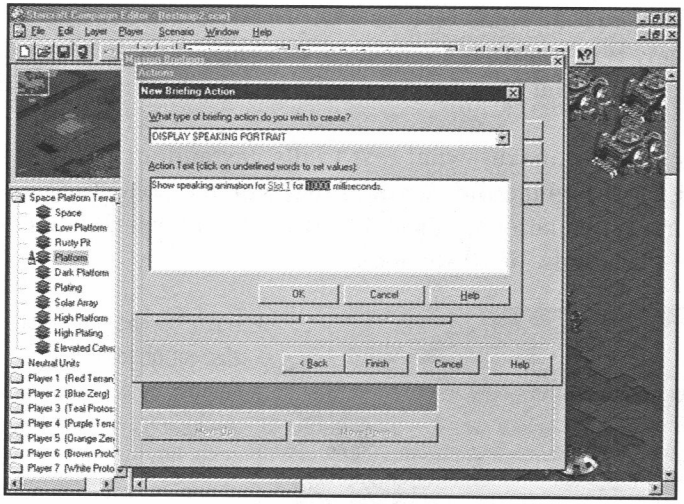


Fig. 5-8. Kerrigan's portrait in Slot 1 will appear animated for 10 seconds.

Tip

A millisecond is one one-thousandth of a second, so each 1,000 milliseconds equals one second. Keep these figures in nice round numbers when you can.

In your briefing, Kerrigan's portrait now will display in the leftmost window and appear to talk for 10 seconds.

There's one more thing you can do with Kerrigan's portrait: make it disappear. Return to the New Briefing Action menu and select Hide Portrait. You have only one option here—which portrait to hide. Click on Slot Number and choose the portrait you wish to make disappear. For now, keep the default. This will cause Kerrigan's portrait to disappear at the end of her 10 second animation.

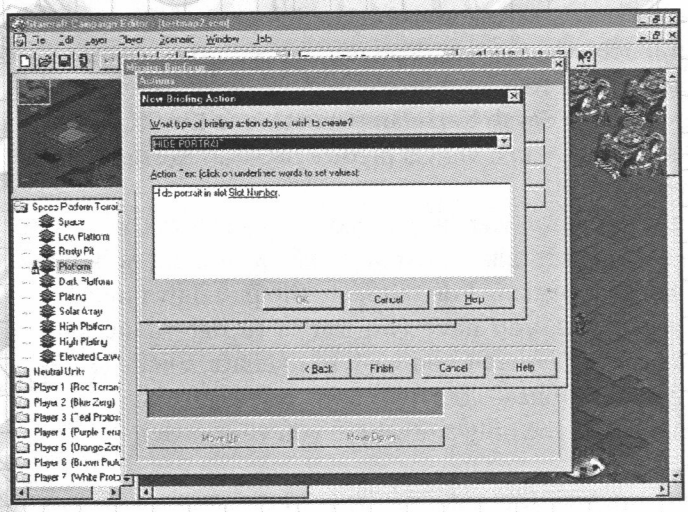


Fig. 5-9. Use the Hide Portrait command to make a portrait disappear from the briefing.

Text Messages

Mission briefings usually include some text, as well. The Text Message action allows you to display any text you'd like in the large central box of the Mission Briefing screen. Again, go to the New Briefing Action menu and select Text Message.

You have two decisions to make—what text to display and for how long. Click on Number first, to determine the length of time you'll display the message. As with the speaking animations, this is determined in milliseconds. For now, choose 10,000 milliseconds, or 10 seconds.

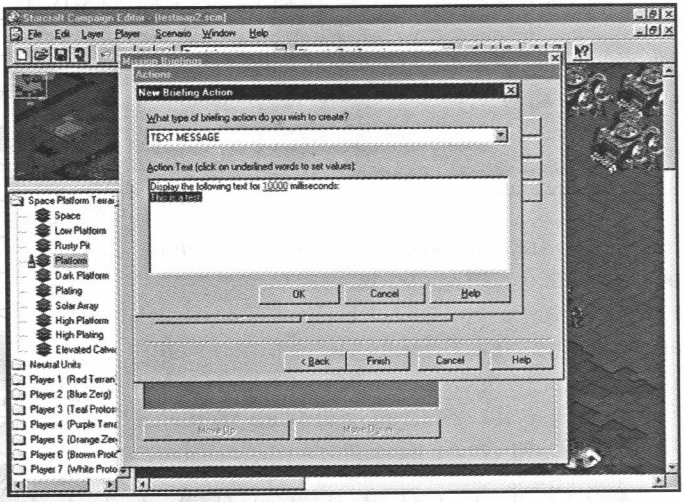


Fig. 5-10. Text is a major element of your Mission Briefing, especially when you have no ready-made sound files to play.

Click on Text. You'll view a large dialogue box in which you may type the text you want to display in the middle box of the briefing. Text that is too long to fit in the box will scroll automatically. For now, type "This is a test" and click OK. You've just told the briefing to display that text in the central window for 10 seconds at the start of the mission.

Playing .Wav Files

To add sound to your briefings, you must use this command. Return to the New Briefing Actions menu and select Play Wav.

If you loaded .wav files earlier to use in your scenario, now you can select the one you wish to play during your briefing. (For more information on loading .wav files, refer to Chapter 2.)

A complicated but far more efficient way to use .wav files with your briefing involves creating a "Transmission." Return to the New Briefing Action menu and select Transmission. This action involves five variables. It combines the actions of displaying a speaking portrait, running text in the central window, and playing a .wav file simultaneously. First, you must choose the portrait you want to animate. (You must have displayed the portrait already using a Show Portrait action.)

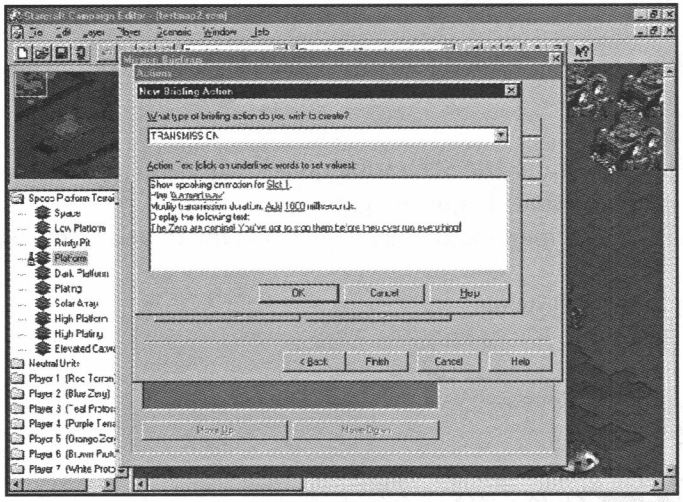


Fig. 5-11. Transmission commands are complex but far more efficient because they combine three actions in one.

Next, determine the .wav file you wish to play, much as you do when simply playing a .wav file. Now you can adjust the way the sound file plays. Click Set to keep the .wav file playing at its normal duration, or to increase or decrease its playing time. Next, click Number to specify the amount of time you want to add or subtract from the sound. Finally, you may include the text you want displayed as the .wav file plays.

Wait Commands

Although this command simply tells the briefing to wait for a specified amount of time, it's among the most important commands you'll issue. When you issue Wait commands correctly, your briefing will run perfectly; without them, your briefing will appear disjointed, or will flash by too quickly to hear or read.

Wait commands tell the system to wait for a specified number of milliseconds. Use a Wait command after every timed command or set of timed commands you use. This will become clearer in the following examples. For now, back out of the briefing you've been creating by canceling it. Now you'll develop the actual briefing you'll use with your scenario.

Sample Mission Briefings: Simple

To prepare a simple but effective briefing for the scenario you started in the last chapter, follow these steps:

1. Click on Mission Briefings, under the Scenarios pull-down menu. Click New, and select Player 1 to indicate that this briefing is for Player 1.
2. Click on New in the Actions menu and select Mission Objectives from the pull-down menu.
3. Click Text and type in the following:

```
Test new Marine specifications.
Destroy all Zerg forces.
```

Then click OK.

4. Click on New in the Actions menu and select Show Portrait from the pull-down menu.
5. Find the portrait of Arcturus Mengsk under Heroes in the Terran Units folder and choose to display him in Slot 4. Click OK.
6. Click on New in the Actions menu and select Display Speaking Portrait from the pull-down menu.
7. Display the portrait in Slot 4 (Mengsk) for 10,000 milliseconds. Click OK.
8. Click on New in the Actions menu and select Text Message from the pull-down menu.
9. Display the following text for 10,000 milliseconds:

```
Commander: Zerg forces have held this pass too long, but
they're dug in firmly and we've been unable to rout them.
We give the job of eliminating them to you. To help you,
we've modified the armor and weaponry of our Marine forces.
You'll find they're more expensive but much tougher in com-
bat. Good luck.
```

Click OK.
10. Click on New in the Actions menu and select Wait from the pull-down menu.
11. Tell the briefing to wait 10,000 milliseconds. Click OK.
12. Click on New in the Actions menu and select Hide Portrait from the pull-down menu.
13. Hide the portrait in Slot 4. Click OK.
14. Click on New in the Actions menu and select Text Message from the pull-down menu.
15. Display the following text for 10,000 milliseconds:

```
END OF BRIEFING
```

Click OK.
16. Click on New in the Actions menu and select Wait from the pull-down menu.
17. Tell the briefing to wait 10,000 milliseconds. Click OK.
18. Click Finish in the Actions menu.
19. Click Close in the Mission Briefings menu.

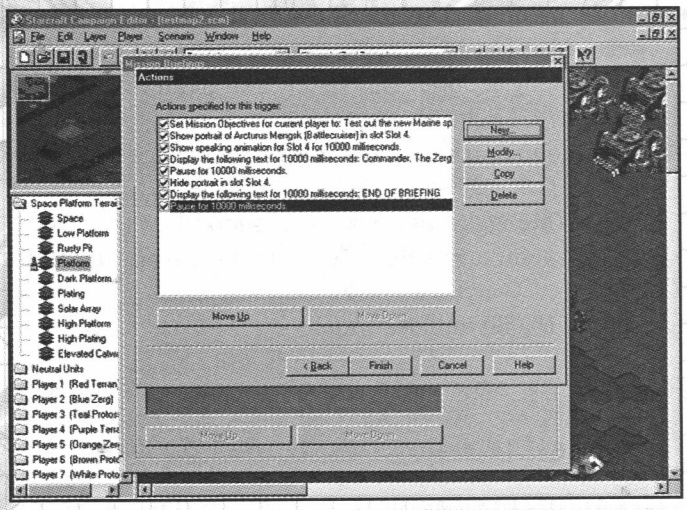


Fig. 5-12. Here's the full set of actions, all ready to go. Now you just need to make sure it works correctly.

Here's what you told StarEdit: When someone starts up this scenario, put a picture of Arcturus Mengsk in the right-hand window and display your Mission Objectives in the lower-left corner. For 10 seconds, Mengsk's portrait will appear to talk. As he does, the large window will show the text. The Wait command will ensure that the system will display the animation and text for 10 seconds before starting the next action. Then the picture of Mengsk will disappear, and text will display for 10 more seconds telling the player that the briefing is over.

Now save your scenario and try it out!

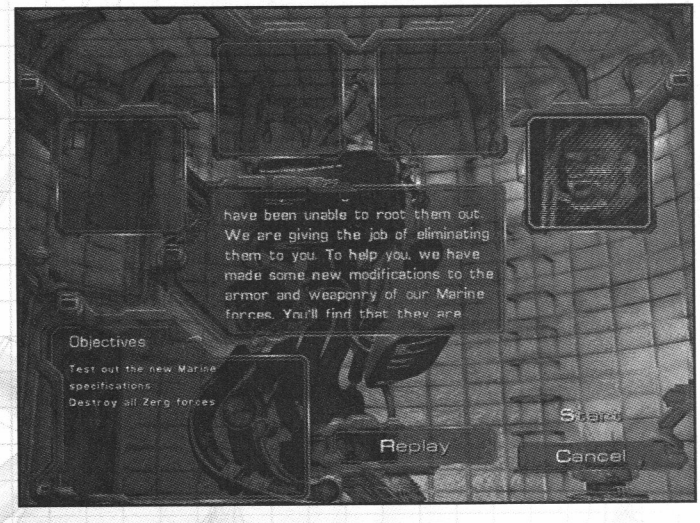


Fig. 5-13. Here's the briefing in action: Mengsk is speaking, he's in the correct slot, and the text is scrolling.

Sample Mission Briefings: A More Complicated Example

The briefing you just created will serve, but it isn't very exciting. Still, if you're happy with it, feel free to keep it.

In this section, you'll use a more complicated procedure to develop a more interesting, more visually appealing briefing. Follow these steps:

1. Select Mission Briefings from the Scenarios pull-down menu. Highlight the briefing you just created and click Delete. (If you don't delete the first briefing, players will view both briefings when they start the mission.) Next, click on New and choose the briefing for Player 1.
2. Click on New in the Actions menu and select Mission Objectives from the pull-down menu.
3. Click Text and type the following:

```
Test new Marine specifications.
Destroy all Zerg forces.
```

4. Click on New in the Actions menu and select Show Portrait from the pull-down menu.
5. Find the portrait of Arcturus Mengsk under Heroes in the Terran Units folder; choose to display him in Slot 1.
6. Click on New in the Actions menu and select Display Speaking Portrait from the pull-down menu.
7. Animate the portrait of Mengsk in Slot 1 for 10,000 milliseconds.
8. Click on New in the Actions menu and select Text Message from the pull-down menu.
9. Display the following text for 10,000 milliseconds:

```
Commander: The Zerg have been holding an important strategic
location for too long. They're well dug-in and have resisted
all efforts to destroy them. You must eliminate them.
```

10. Click on New in the Actions menu and select Wait from the pull-down menu.
11. Pause the briefing for 12,000 milliseconds. This not only will cause the briefing to wait while the text scrolls, but it will add an extra two seconds of pause before the next action occurs.
12. Click on New in the Actions menu and select Show Portrait from the pull-down menu.

13. Select the portrait of the Terran Marine from the menu and display it in Slot 3.
14. Click on New in the Actions menu and select Display Speaking Portrait from the pull-down menu.
15. Display the animated portrait of Mengsk in Slot 1 for 10,000 milliseconds.
16. Click on New in the Actions menu and select Text Message from the pull-down menu.
17. Display the following text for 10,000 milliseconds:

> To assist you in battle, we've made significant upgrades to our Marines. You'll find them far more effective in battle, but they're also more costly to build. We hope they'll carry the day for you.

18. Click on New in the Actions menu and select Wait from the pull-down menu.
19. Pause the briefing for 12,000 milliseconds.
20. Click on New in the Actions menu and select Hide Portrait from the pull-down menu.
21. Hide the picture of the Marine unit in Slot 3.
22. Click on New in the Actions menu and select Show Portrait from the pull-down menu.
23. Select the Terran Dropship portrait from the menu on the left and display it in Slot 4.
24. Click on New in the Actions menu and select Display Speaking Portrait from the pull-down menu.
25. Animate the portrait of the Dropship in Slot 4 for 5,000 milliseconds.
26. Click on New in the Actions menu and select Text Message from the pull-down menu.
27. Display the following text for 5,000 milliseconds:

> We're ready to go whenever you are, Commander.

28. Click on New in the Actions menu and select Wait from the pull-down menu.
29. Pause the briefing for 7,500 milliseconds.
30. Click on New in the Actions menu and select Hide Portrait from the pull-down menu.
31. Hide the portrait of Mengsk in Slot 1.
32. Click on New in the Actions menu and select Hide Portrait from the pull-down menu.

33. Hide the portrait of the Dropship in Slot 4.
34. Click on New in the Actions menu and select Text Message from the pull-down menu.
35. Display the following text for 10,000 milliseconds:

```
END OF BRIEFING
```

36. Click on New in the Actions menu and select Wait from the pull-down menu.
37. Pause the briefing for 10,000 milliseconds.
38. Click Finish in the Actions menu.
39. Click Close in the Mission Briefings menu.

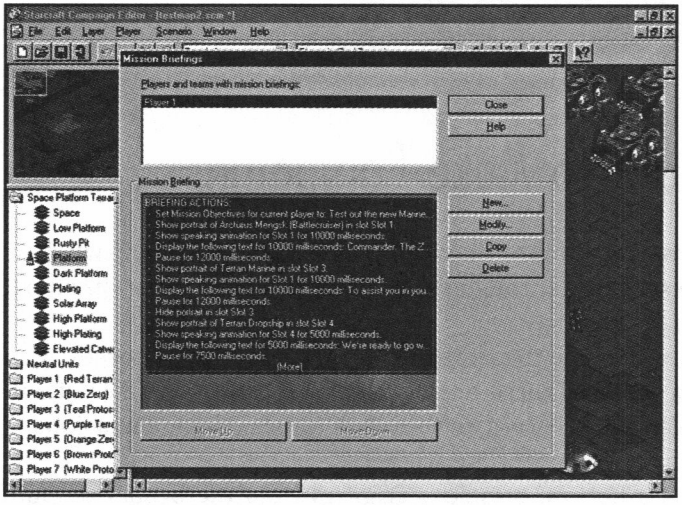


Fig. 5-14. Your new mission briefing is too long to appear in one box, but it will work as planned.

Whew! This was time-consuming, but the results should be worth it. Here's what you did: Again, your objectives appear in the lower-left corner. This time, Mengsk appears in the left window and gives the player a little information. Then, a picture of a Marine appears, and Mengsk provides information about the new unit. Next, the portrait of the Marine disappears, a Dropship appears, and the pilot tells the player that everything is ready. Finally, both the Dropship and Mengsk disappear and the player views a message saying the briefing is over.

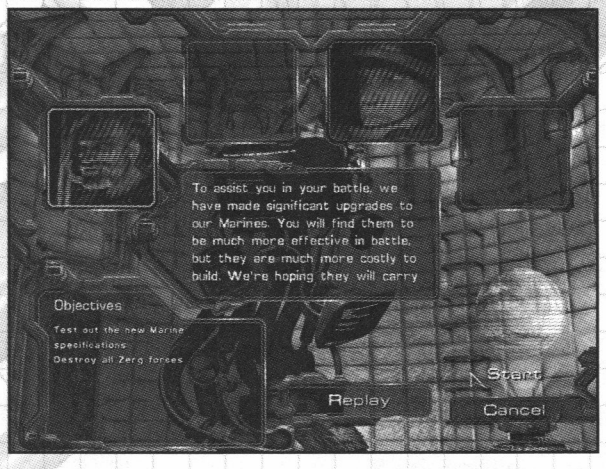


Fig. 5-15. And here's your briefing in progress. It gives the player all necessary preliminary scenario information and is interesting to watch.

Linking Missions to Make Campaigns

At times you'll want to create more than one mission based on a particular theme, and to have players go through them in a particular order. In short, you'll want to develop a campaign.

For instance, if you like the idea of using beefed-up Marine forces, you might want to create a series of three or four missions using the new units in different situations, against both Zerg and Protoss forces. Linking missions is simply a matter of using the appropriate trigger.

Click on the Triggers button, or select Triggers from the Scenarios pull-down menu. You'll notice that a few triggers are in place already. Ignore these for now. You also will notice that this looks quite a bit like the menu you used to create your Mission Briefing.

Essentially, linking missions to make campaigns is an action you tell the game to take when a player wins your scenario. Instead of simply telling the player he or she has won, you'll also load the next challenge.

In the Triggers menu, click on New, select Player 1, and hit Next. This does the same thing as starting a Mission Briefing does. The condition you're about to generate will affect Player 1, your human player.

This will bring up the next menu, the Conditions list. This sets the conditions necessary to cause the game to load the next scenario. Click on New to get a New Conditions dialogue box (identical to the New Briefing Action box). Click on the pull-down menu and select Control. Now click on Player, and select Player 2, the blue Zerg forces—your computer opponent.

Next, click on Quantity and select Zero or Fewer. Leave the number alone (the default is zero) in the lower left. Finally, click on Unit and choose any unit. Then hit OK to return to the Conditions dialogue box.

You've told the game that the trigger you're going to create will happen when the computer opponent (Player 2) has 0 or fewer buildings and units—that is, when all Zerg have been wiped out.

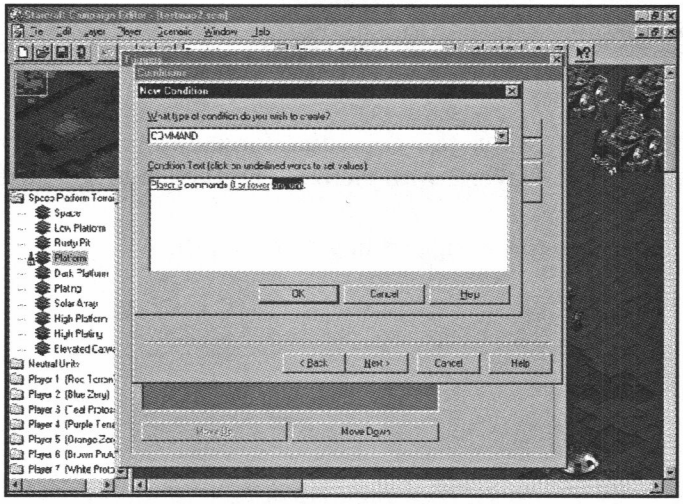


Fig. 5-16. Setting conditions tells the game when to activate the trigger.

Now you must tell the game what to do. Click Next in the Conditions dialogue box to proceed to the final section of setting this trigger. You'll view the Actions dialogue box (identical to the Conditions box). Click New to go to the New Actions box. Select Set Next Scenario from the pull-down menu. Now click on Scenario in the dialogue box, select any scenario, and hit OK. Click New again and scroll down to Victory, and hit OK. Then click on Finish to set the trigger and close the Trigger menu.

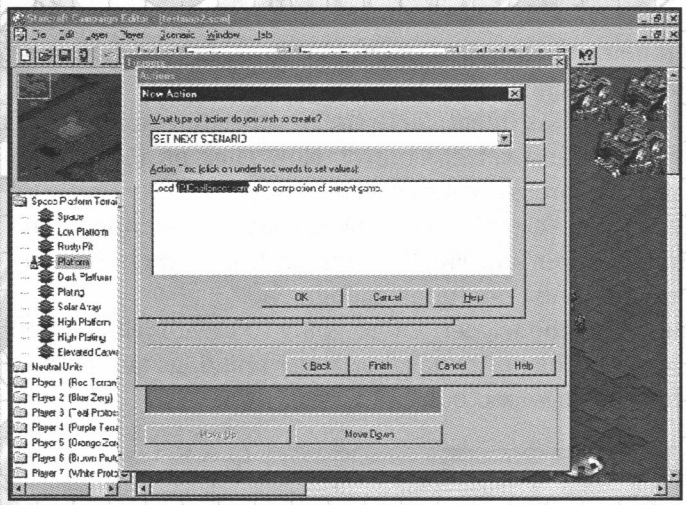


Fig. 5-17. Now, when the player wins your scenario, the next mission will load automatically.

You've told the game that, when the computer forces are completely destroyed, it should reward the Player 1 with victory, and then load the next scenario.

The following two chapters cover triggers in greater depth.

Note

Not all your settings will transfer from scenario to scenario. If you load another mission after the one you created, the Terran Marines will return to their original specifications. You must effect those values in every scenario in the campaign to keep the Marines beefed up.

Chapter 6
Trigger Basics

A trigger is something special that happens during the game as the result of a particular event or the attainment of a goal. For instance, when a player destroys all the enemy's buildings and units (attains the goal), the scenario ends in victory for the player (the special happening that results). Without triggers, all scenarios effectively would be the same. And they would never end: there would be no conditions for victory or defeat.

Triggers work by combining conditions and actions. When a player meets a particular condition or set of conditions, a particular action takes place. Conditions can be virtually anything, from bringing a certain unit to a specified location to wiping out all enemy structures. Actions can range from something as simple as playing a .wav file to something as complex as creating several dozen enemy units and having them swarm the player.

This chapter looks at the various possible conditions and actions, and provides a few examples.

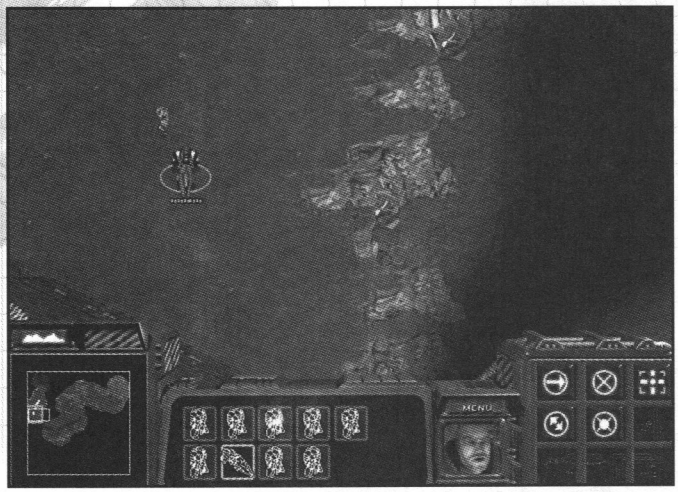


Fig. 6-1. Triggers control virtually everything in a scenario, from the pings on the Mini Map and the player's starting resources to Victory conditions and the playing of .wav files.

Definitions

Action: The event that occurs when trigger conditions are met.
AI Script: A group of specific actions a computer opponent will use.
Condition: The event or events that must take place to cause a trigger to become active.
Current Player: The player the trigger affects. Conditions can be created so that when one player does something, an action occurs to another player. Other conditions can affect only the current player.
Duration: The length of time a trigger occurs.
Label: The name of a leader board or other scoreboard.
Location: A specific map area defined using the Location layer when building a map.
Player: Any human or computer player, or any force as defined in the forces menu.
Switch: Allows you to determine the number of times the trigger will become active.
Trigger: A combination of conditions and actions that generates a particular event at a specific time. Must include at least one player, one condition, and one action.
Unit: Anything a player can produce in a *Starcraft* scenario. Includes all units and all structures.

Conditions

Certain conditions must be met before an action can take place. Most conditions can be as general or as specific as you wish. This is where you can really personalize your scenarios. Interesting trigger combinations are what make missions different and interesting.

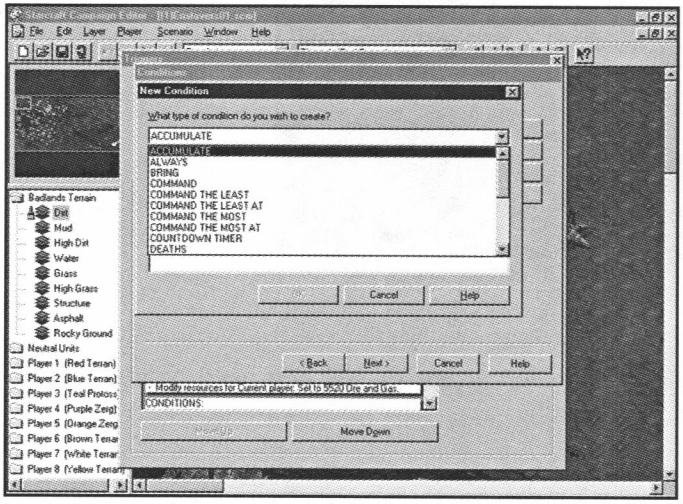


Fig. 6-2. Conditions tell the game when to set off a trigger.

Almost every condition involves defining certain variables—usually a player or force and another variable, such as a location. Each trigger must have at least one condition, and it's possible for a trigger to have two, three, or a dozen conditions. In fact, there is no limit. All conditions must be met before a trigger's actions can take place.

Resource Conditions

The following conditions center around the accumulation of resources—minerals, gas, or both.

ACCUMULATE

An Accumulate condition requires a player to gather a given amount of resources, either minerals, gas or both. To use this condition, you must define the player or force, the resource or resources gathered, and the amount required.

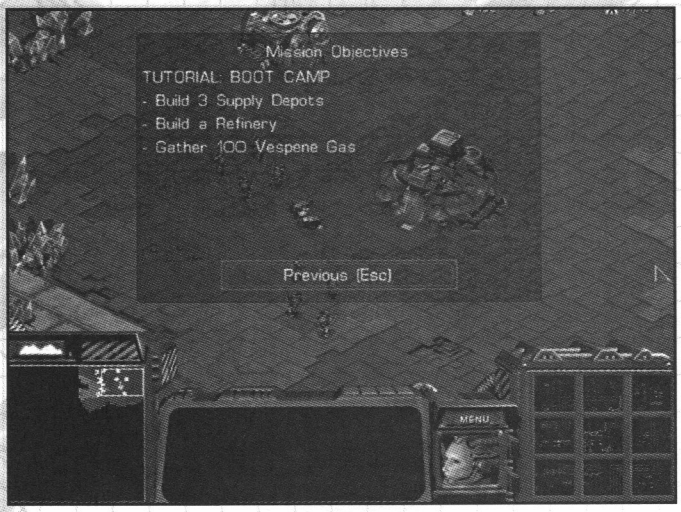


Fig 6-3. The tutorial for *Starcraft*'s Terran campaign contains an Accumulate requirement: the player must harvest 100 units of gas to win.

Least Resources

This condition can affect only the current player. You need only define the resource or resources in question.

Most Resources

The opposite of Least Resources, this condition affects the current player only. As with Least Resources, you need only define the resource or resources.

Always/Never Conditions

Use Always and Never conditions to produce a continual effect or to prevent an event from occurring.

Always

An Always condition keeps a trigger continually active. For example, you may want a computer opponent always to act a certain way.

Never

Never conditions prevent an event from occurring. Use it, for instance, to prevent a computer opponent from moving troops into a certain area.

Bring Conditions

Use these conditions to have a player bring a particular item, unit, or group of units to a specific location. Any combination of units is possible, including Heroes. You'll find many examples of Bring conditions in *Starcraft*'s three campaigns. Often, these conditions will determine victory or generate a major turning point. You must define the player or force, unit type and quantity, and the location the units must reach.

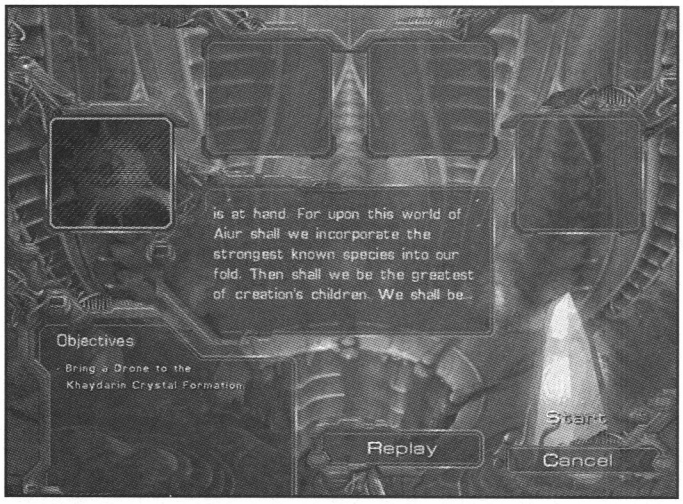


Fig. 6-4. The ninth Zerg scenario centers around bringing the Khaydarin Crystal to a particular location.

Tip

You can define locations anywhere on your map using the Location layer. Refer to Chapter 2 for details.

Command Conditions

These conditions become active once the player has a certain number of units in a general or specific location.

COMMAND

This most basic Command condition requires you to define the player and the type and quantity of unit affected. You may specify any type of unit or structures, or combination thereof. An excellent example of this occurs in the second mission in *Starcraft*'s Terran campaign, where the player must create 10 Marines to achieve victory.

Command the Least

This condition, defined for the current player, activates the trigger when the player commands the fewest of a given unit type or structure. You need define only unit type.

Command the Least At

This adds an extra variable to a Command the Least Condition. In this case, the trigger becomes active when the defined player commands the fewest of a type of unit or structure in a given location.

Command the Most

The opposite of Command the Least, this, too, is defined for only the current player. It activates the trigger when the player commands the most of a certain type of unit or structure. You need define only the type of unit.

Command the Most At

This adds a location variable, which you must define, to the Command the Most condition.

Time Conditions

These conditions generate specific events at specific times during the scenario. Use them to have events occur at a specific time after the start of the scenario, or in a countdown timer.

Countdown Timer

This condition is met at a specific time on a countdown timer included within a mission. For example, you may wish the computer opponent to mount a massive attack a minute or two before the timer ends, forcing the player to defend against a last-ditch attack from the enemy. You need define only the time of the event.

Fig. 6-5. This massive Zerg attack takes place when 90 seconds remain on the timer.

Elapsed Time

This condition generates an event after a specific number of seconds of gameplay. Again, you need define only the time of the event.

Kills Conditions

These conditions generate an event when a player or force inflicts or suffers a specific number of kills.

Deaths

This condition activates a trigger after a player or force suffers a specific number of casualties. You must define the player, number of deaths, and unit type.

Kill

Triggers with this condition become active after a given player kills or destroys a specific number of units or structures. You must define the player, number of kills, and unit type or structure.

Least Kills

This condition is most useful for multiplayer scenarios, either to assist or to penalize a player for having the fewest kills of a given unit type or structure. This condition activates only for the current player, and you must define the unit or structure.

Most Kills

This condition can affect only the current player. It comes into play once the player achieves the most kills of a given unit or structure. You need define only the type of unit to be killed.

Opponents

Triggers with this condition activate when a given player has a specific number of opponents left in the scenario. You must define the player and the number of opponents.

Score Conditions

You may define scores a number of ways in a scenario, including number of kills, buildings razed, and units or buildings created.

Highest Score

Limited to the current player, a trigger with this condition activates when the player has the highest score. You need define only the type of score.

Lowest Score

This comes into play when the current player has the lowest defined score. Again, you need define only what the score comprises.

Score

Triggers with this condition activate once a given player achieves a specific score. You must define the player, score total, and what the score comprises.

Switches

You may define conditions as met when a given switch is on or off. The "Actions" section, later in this chapter, provides details on switches.

Trigger Actions

Trigger Actions take place once a player meets a given condition or set of conditions. These can be simple occurrences, such as the playing of a .wav file, or complicated ones, such as adding resources and units to a computer player's army, or generating an attack. As with conditions, you may incorporate numerous actions into a particular trigger.

Mission Actions

These actions generally take place at the beginning of a scenario, but you may include them at certain points later in the game. In general, these control what the human player or players must achieve, what they start with, and what happens when the scenario ends.

Set Mission Objectives

This action changes the Mission Objectives the player views. Using this action, you can generate an additional requirement. This doesn't change Victory conditions, only the player's awareness of them. This way, you can alter the objectives partway through your scenario.

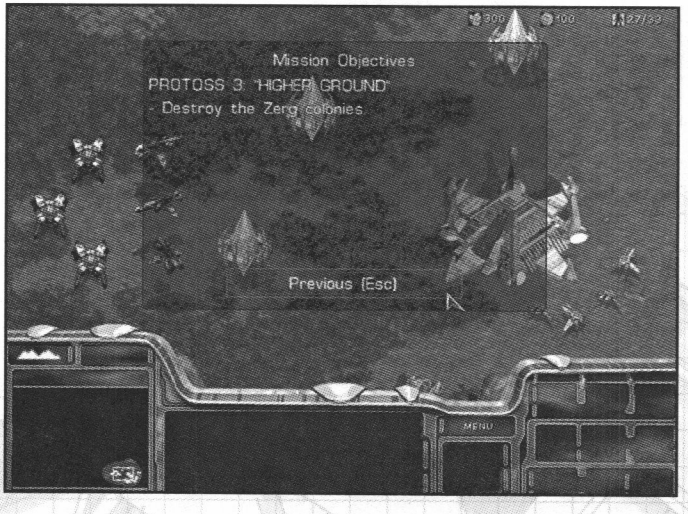


Fig. 6-6. Use this action to change the Mission Objectives.

Set Next Scenario

This action allows you to create campaigns by linking scenarios. For more on creating campaigns, refer to Chapter 5.

Set Resources

This action changes the resource amount a particular player owns. Usually you'll use this at the beginning of a scenario to adjust a player's resource totals from their defaults. You also may use it to add or subtract resources from a player's total once that player meets certain conditions. You must define the player and the quantity and type of resource affected.

Set Score

This action allows you to set a player's score and add or subtract points. You must define the player and the score total or number of points to add or subtract.

Victory/Defeat Actions

These actions define how a scenario is won or lost. The default Victory condition for all players is the destruction of all enemy buildings, but you may change Victory and Defeat conditions to whatever you wish.

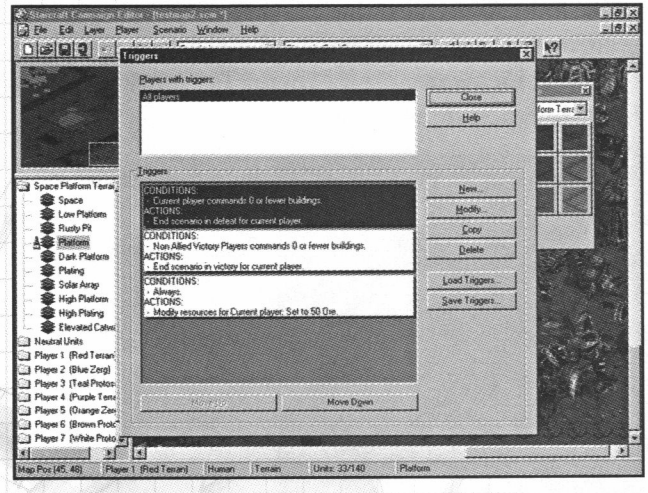


Fig. 6-7. Even the most rudimentary scenario must have Victory and Defeat conditions.

Defeat

When the player meets the trigger's conditions, the scenario ends in defeat. You may define this for only the current player.

VICTORY

Likewise, when the player meets the conditions of a trigger with this action, that player wins the scenario. You may define this for only the current player. You must include a Victory Action before a player can win a scenario.

AI Actions

These control the way the computer opponents play in your scenario. Using AI scripts, you can make computer opponents easy, difficult, or nearly impossible to beat. You can force them to attack in suicide runs, or turn them from enemies to allies. There are many AI types.

RUN AI SCRIPT

This AI Script action is limited, and affects the given computer-controlled army across the board. You have three choices when running an AI script:

Send All Units on Strategic Suicide Missions: This AI script causes all computer opponent troops to mass and attack one unit or structure at a time until they or the human player's forces are wiped out.
Send All Units on Random Suicide Missions: The computer opponent's units will attack and continue attacking enemy forces at random until killed.
Switch Computer Player to Rescuable: The computer player's units and structures become rescuable forces instead of foes.

Fig. 6-8. Without an AI script, computer enemies such as these Zerg will act only defensively.

Run AI Script at Location

While it seems more limited at first, this is a far more useful set of AI scripts than the foregoing. This action allows you to run a given AI script at a defined location on the map. You may choose from many possibilities; for all but "Value this Area Higher" and "Enter Closest Bunker," you must define the type of force the computer opponent will build.

Custom Level: Use Custom Level AI in stand-alone scenarios to provide the greatest possible challenge. Custom Level AIs will build their town, expand aggressively, and attack any enemy they encounter.
Campaign Easy: Campaign Level AI is designed for the human player to beat eventually. Until destroyed, campaign AIs seek to wear the player down over time. The Easy level of AI builds only basic units and doesn't explore the research tree.
Campaign Medium: Computer opponents with this level of AI will build basic units and some advanced ones and will investigate initial parts of the research tree.
Campaign Hard: The most difficult Campaign AI, this level will build all available units and research all upgrades and technologies.
Campaign Area Town: This is purely a defensive AI that concentrates on resource-gathering and defense.
Value This Area Higher: The AI will value the defined location; when this AI becomes active, the computer opponent will send units to control and defend the defined area.
Enter Nearest Bunker: This tells Terran ground units to enter the closest available open Bunker (affects only Terran AIs).

Note

Here are a few notes on using AIs:

- Except for using Town AIs to create computer-controlled mining bases, you should stick with one Campaign or Custom AI for a scenario.
- Custom AIs are designed to start with the same level of resources the human player does.
- Campaign Level AIs do better when given an initial resource or unit advantage (or both). 4,000–6,000 minerals and gas are recommended as starting levels for Campaign AIs.
- Campaign AIs must be run in locations that have both minerals and gas—its base (and perhaps secondary base) location, for instance.

Setting Switches

This action allows you to set or unset a switch, or change the setting of a given switch, to control a variety of actions.

View Actions

These actions allow you to move the player's view to a place on the map where a particular action is taking place.

Center View

This action centers the player's view at a defined location. It effectively pauses any other action in the trigger until the player's view is centered in that spot. (This doesn't affect computer players.)

Mini Map Ping

This action creates a "ping," or a highlighted burst, on the player's Mini Map at a specified location. You can use this to call the player's attention to a particular area.

Move Location

This action allows you to change a specified location on a particular unit or units. You must define the location, the player, and the units. You also must define the map area where the location will search for units. The new location area will center on the defined unit or units. You can use this action, for example, to cause computer enemies to attack a particular unit or unit type.

Move Unit

Use this action to move units from one location to another—effectively teleporting it across the map. You must define the affected player, the units to transport, and both starting and destination points.

Note

Switches allow you to use and reuse triggers, or to set up a series of actions that culminate in one large, all-encompassing action. For instance, perhaps you want something to occur when the player meets a certain number of goals. You could create a switch for each goal—one sets when the player builds his tenth building, another after the player creates his or her 25th unit, a third when a second Command Center is built, and a fourth after the player accumulates 20 kills. Then, once all the switches are set, a significant action would occur, such as a major attack by the computer opponent. But until the player has set every switch, the attack won't occur.

Fig. 6-9. Use the Move Unit action to teleport units across the map.

Create Actions

You can create any type of unit or building from thin air using this action.

Create Unit

When your conditions are met, this action creates new units or structures. Generate large attacking enemy groups instantly at a specific location. You must use a separate Create Action for each unit you create, and must specify unit type, location, and the player the unit belongs to.

Create Unit with Properties

Use this action to develop a unit or structure with any properties you desire.

Text and Sound Actions

These actions generate text or .wav file messages at specific times. Often, you'll combine these with a Mini Map Ping, Center View, and other actions.

Display Text Message

Displaying a text message gives the player information at a specific time. The message is active for only the current player, and you need create only the text you wish to display.

Mute Unit Speech

Use this action to mute the sounds of other units and battles in the game when you play a .wav file. The sound of any unit not involved with the trigger will be muted for the current player.

Play Wav

When specific conditions are met, this command plays a particular .wav file.

Talking Portrait

This action generates an animated portrait of a given unit in the lower-right corner of the player's screen. Usually this is combined with a .wav file or a text message.

Transmission

In effect, this action displays a talking portrait, plays a .wav file, and shows a text message for the current player.

Fig. 6-10. The Transmission action allows you to display a portrait and run a .wav file at the same time.

Unmute Unit Speech

Return muted sound to its normal volume by using this action. Make sure you unmute unit speech after your .wav file plays.

Unit Actions

These actions control various unit and structure characteristics.

Kill Unit

When the conditions for this action are met, the specified units or structures will be destroyed. You must define the units to be destroyed and the player who will be the victim.

Kill Unit at Location

This action adds another variable to the Kill Unit action. All specified units or structures for the player you define will be destroyed at a given location.

Remove Unit

Similar to the Kill Unit action, this simply removes the unit or units in question from the game without killing them. This can be important in scenarios where kills are being counted for a particular reason.

Remove Unit at Location

Similar to Remove Unit, this action removes a unit or units at the location of your choice.

Set Doodad State

Some Doodads, such as doors and hidden turrets, have several different states. For instance, doors can be open or closed; gun turrets or wall-mounted guns can be active or inactive. This action allows you to change the Doodad's state, opening doors or causing hidden turrets to pop up and attack. You must draw a Location box around the chosen Doodads to enable them.

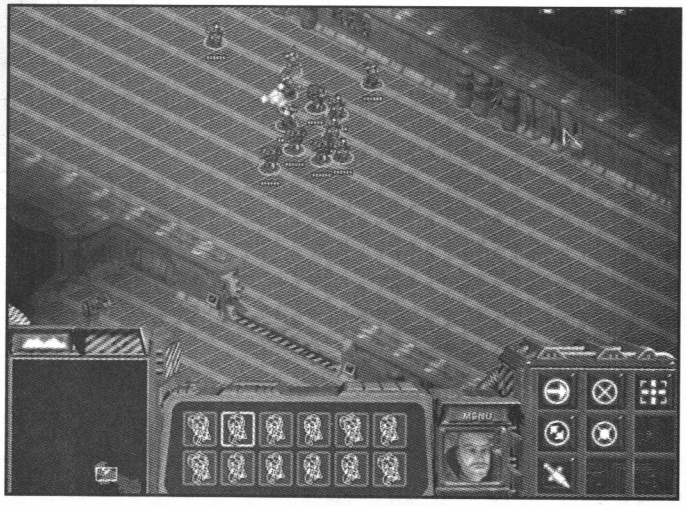


Fig. 6-11. The Doodad state of the door and wall-mounted rocket launcher changed when the Marines moved close to them.

Tip

An enabled door is closed. An enabled turret or wall-mounted gun is active and able to attack.

Set Invincibility

This action allows you to render a unit or structure invincible, or turn off the invincibility of a previously invincible unit. Invincible units can't be targeted or attacked, and take no damage. You must define the player, the chosen unit or units, where the action takes place, and the state of invincibility (whether the unit becomes invincible or can be damaged).

Leader Board Actions

Leader Board actions allow you to display a leader board for all players in the scenario. You can name your leader board anything you like.

There are two types of Leader Board actions—normal leader boards and leader board goals. A Leader Board Goal action shows how many points or kills, or how much of a given resource or control it takes to achieve victory in terms of how much remains outstanding. The lower the number, the closer the player is to reaching the goal.

Leader Board (Control at Location)

This action shows a leader board of control for a particular unit at a specified location.

Leader Board (Control)

This action shows a leader board of control for a particular unit anywhere in the scenario.

Leader Board (Greed)

This action shows a leader board that lists players in order of how close they are to gathering a particular amount of a given resource.

Leader Board (Kills)

This action displays a leader board showing kills, either general or for a particular unit or structure type.

Leader Board (Points)

This action displays a leader board showing how many points the player or players in the scenario have scored.

Leader Board (Resources)

This action displays a leader board showing the amount of a particular resource players have gathered.

Leader Board Goal (Control at Location)

This action shows a leader board of the players in the order of how close they are to reaching the particular goal of controlling a particular type of unit at a specified location.

Leader Board Goal (Control)

This action shows a leader board of players in order of how close each is to reaching the goal of controlling a particular type of unit anywhere on the map.

Leader Board Goal (Kills)

This action displays a leader board showing players in order of how close each is to reaching the goal of kills for a specified unit or unit type.

Leader Board Goal (Points)

This action displays a leader board showing players in order of how close each is to scoring the scenario's point goal.

Leader Board Goal (Resources)

This action displays a leader board showing players in order of how close each is to mining the scenario's resource goal.

Leader Board Computer Players

This action allows you to determine whether to include the computer players, rescuable players, and neutral players in the leader board. By default, all computer players are included.

Pause Actions

These actions allow you to pause or unpause the game while other important actions take place.

Pause

This pauses the game while other actions in the trigger take place.

Unpause

This unpauses the game after it's been paused. Always include an Unpause command after you've paused the game, or the action won't continue.

Wait

This action causes the trigger's other actions to wait for a given amount of time before occurring. When a Wait action is executed, all other actions in the trigger will be held up, including Center View and Transmission actions.

Preserve Trigger Actions

Triggers, by default, are one-time events. If you want a trigger to activate more than once in a scenario, you must preserve the trigger using this action. A preserved trigger will operate each time conditions are met.

Setting Countdown Timers

This action generates a countdown timer or alters the amount of time the timer stays in effect.

Sample Triggers

Using your scenario from the last chapter, first return to your map and go to the Location layer. Create a location that encompasses the entire Terran base, one that incorporates the central platform, and one that incorporates the Zerg base. Place a final location in the upper-right corner of the map. Now click on the Triggers button, or select it from the Scenarios pull-down menu.

Three triggers are in place already: one controls defeat (when a player has no buildings), one controls victory (when a player's opponents have no buildings), and one controls starting resources (players start with 50 minerals). Delete these.

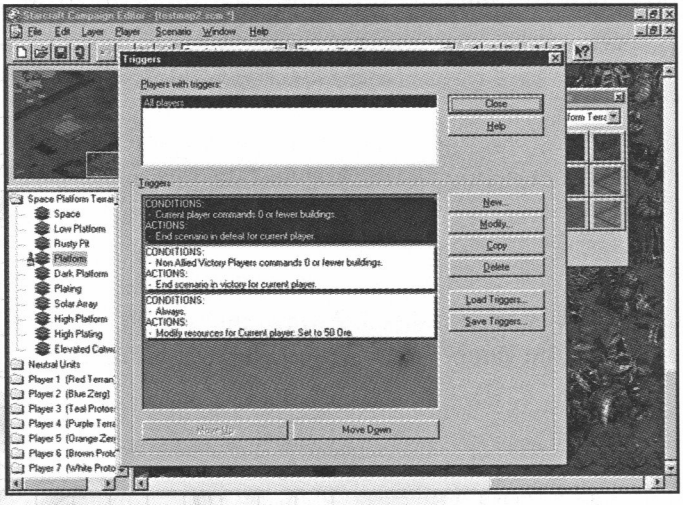


Fig. 6-12. Get rid of the default triggers. You'll add your own soon enough.

Now you must decide how you want the scenario to unfold. In the last chapter, you told Player 1 that the objectives were to test the new Marines, start a base on the central platform, and destroy the Zerg. You'll need to incorporate those into your Victory conditions for Player 1. Obviously, you'll want an AI for the computer player. And you should change the starting values of minerals and gas for both players.

To make things more interesting, add a Zerg rush against the player. With those in mind, you can create some triggers to run your scenario. Take the following steps:

1. In the Triggers menu, click New and select Player 1. Hit Next to get to the Conditions menu; hit New again.
2. Select Command and set the trigger for Player 2 to command zero or fewer units of any type. Hit OK.
3. Hit New again, and select Command the Most At. Set this to command the most Command Centers at the location number for the central platform. Hit OK, and then hit Next in the Conditions menu to go to the Actions menu.
4. Hit New and select Victory. Hit OK, and then Finish. You've set the Victory conditions for Player 1: the player must destroy all Zerg buildings and units and build at least one Command Center on the central platform.
5. Back in the Triggers menu, hit New. Select all players and hit Next to go to the Conditions menu. Hit New.
6. Select Command and set the trigger for current player to command zero or fewer units of any type. Hit OK, and then Next to go to the Actions menu.
7. Hit New and select Defeat. Hit OK; then hit Finish. You've set Defeat conditions for both the human player and the computer player. (The computer opponent will be considered defeated when it has no units or structures, but the human player won't win the scenario until he or she builds a Command Center on the central platform.)
8. In the Triggers menu, hit New and select Player 2, the computer opponent. Hit New to go to the Conditions menu.
9. Select Always and hit Next to go to the Actions menu.
10. Hit New in the Actions menu and select Run AI Script at Location. Run the Zerg medium campaign AI at the location of the Zerg base. Hit OK, and then Finish.
11. Hit New in the Triggers menu and select Player 1. In the Conditions menu, hit New and select Always. Then hit OK and Next to go to the Actions menu.
12. Hit New and select Set Resources. Set the ore resources of Player 1 to 500, and then hit OK. Hit New again and select Set Resources. Set Player 1's gas resources to 200. Hit OK, and then hit Finish. Now the human player will start the scenario with 500 minerals and 200 gas.

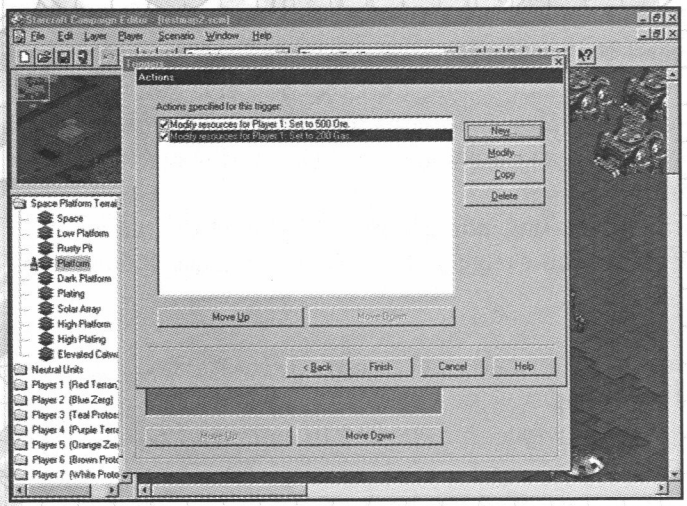


Fig. 6-13. Resources for Player 1 are now set at the beginning of the mission. This way, at least the human player starts with enough to create a few units.

13. Hit New in the Triggers menu and select Player 2. In the Conditions menu, hit New and select Always. Then hit OK and Next to go to the Actions menu.
14. Hit New and select Set Resources. Set the ore and gas resources for Player 2 to 5,000. Hit OK, and then Finish. The Zerg opponent now starts with 5,000 minerals and 5,000 gas.
15. Hit New in the Triggers menu and select Player 2. Hit New in the Conditions menu and select Command. Set the Condition to Player 1 commanding zero or fewer units of any type. Click OK, and then Next.
16. In the Actions menu, select Victory, and then hit OK and Finish. You've just created Victory conditions for the computer player.
17. In the Triggers menu, hit New and select all players. In the Conditions menu, select Command. Set the condition to Player 1 commanding 75 or more men.
18. Hit New and choose Create Unit. Create a Mutalisk belonging to Player 2 in the location number of the upper-right corner of the screen. Hit OK. Click on the action and copy it nine times: you just created a force of 10 Mutalisks at that location that appear when the human player amasses a force of 75 men or more. Hit Finish. Close the Triggers menu.

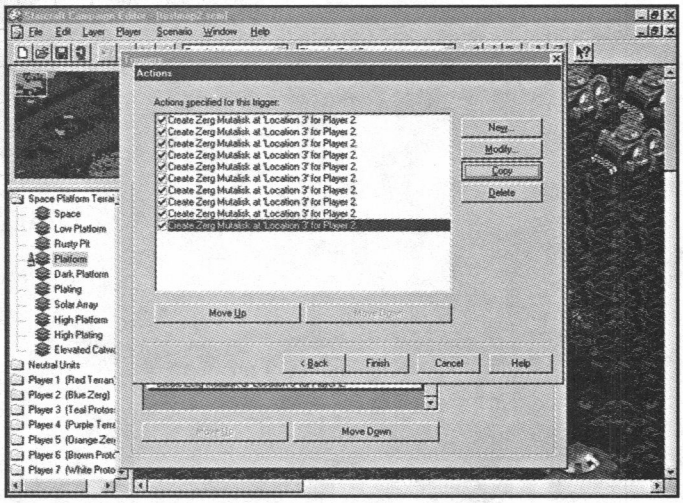
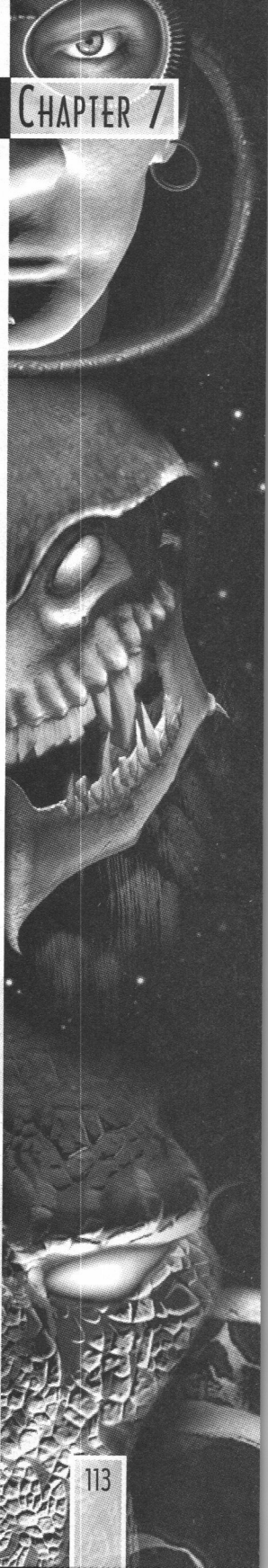


Fig. 6-14. The new Mutalisk force will appear when the player controls 75 or more men.

Now playtest your scenario!

Fig. 6-15. The scenario in action: Zerg forces haven't expanded, but are building units and harvesting actively.

Chapter 7
Using Triggers to Create Unique Scenarios



Each trigger you use in your scenario is important. Those that control victory conditions and the AI are crucial, in fact. Others, such as those that control the timing of enemy attacks or the behavior of neutral armies, give the scenario its basic feel and character. Even those that ping the Mini Map or play a simple .wav file add flavor and character to the overall experience.

You should start thinking about your triggers not as individual commands that produce particular events in the scenario, but as an interlocking *system* of events that gives the scenario its character. The whole of the trigger system is far greater than the sum of its component triggers. And each trigger you add affects the mission's character.

Using Triggers

Victory Triggers

The most important triggers in the scenario by far are those controlling victory and defeat conditions for the various Players. Without them, the mission can't even end. Fortunately, if you don't want to worry about these, you don't have to: a default victory condition—destroying all enemy structures—is included every time you create a new mission.

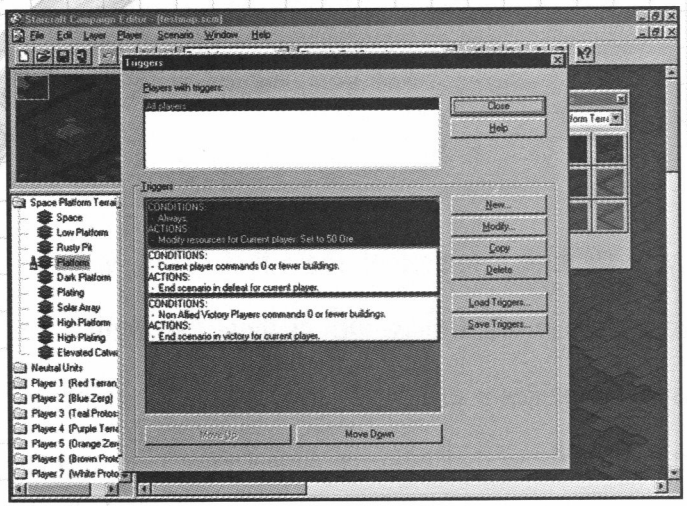


Fig. 7-1. A default victory condition is included with every scenario as soon as you create it.

However, you'll certainly want to change these conditions, or at least add to them, as you develop many of your scenarios. Striving only to destroy enemy structures to win gets boring after a while. This is the other reason these triggers are so important to your scenarios. Victory conditions give a mission its focus. These triggers form the skeleton that the rest of the scenario, with its other triggers, hangs on.

You must remember a few important guidelines as you develop your mission's victory conditions. First, keep them simple. Forcing the player to achieve a half-dozen victory conditions becomes confusing. Feel free to create complicated missions, but create them so the player receives the goals in sequence: meeting one victory condition generates the next. This way, the player need concentrate on only a few things, rather than juggle numerous different goals at the same time.

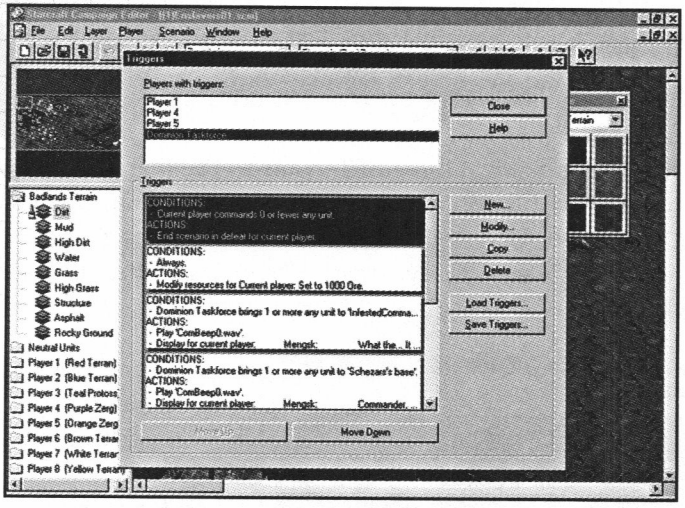


Fig. 7-2. Even a complicated and difficult scenario doesn't need complex victory and defeat conditions.

Second, make your victory conditions consistent with each other and with the scenario's other aspects. Asking your players to destroy one group of Protoss forces without damaging another Protoss force creates difficulty and conflict. Asking them to destroy a fully-functional enemy base without enabling them to create a powerful force of their own makes for frustrated players. Make sure your victory conditions make sense.

Secondary Triggers

Triggers that affect gameplay without affecting victory conditions are called secondary triggers. Although they're not critical, they're still important to your scenario. These are what make your scenario really interesting.

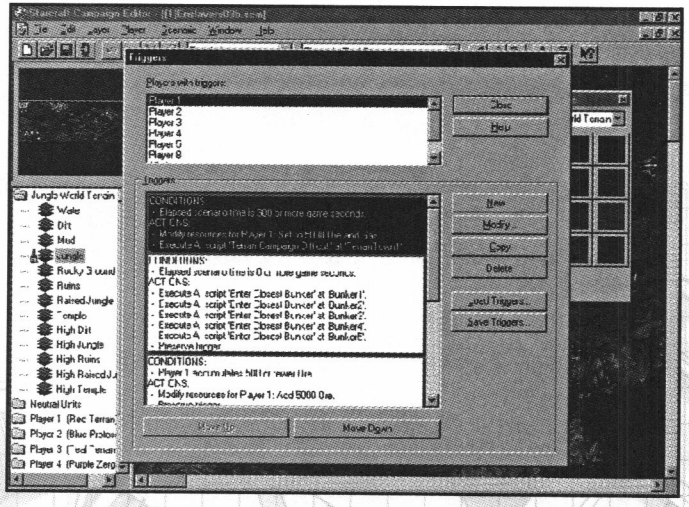


Fig. 7-3. Secondary triggers, for a computer opponent's AI and resources, control much of the way this scenario plays, regardless of victory conditions.

Consider the scenario you created in the last chapter. It was a fairly standard scenario, pitting a human player with a Terran Force against a computer-controlled Zerg army. Victory conditions on both sides were to destroy all enemy structures and units. Even keeping the same conditions, you can do a lot of interesting things with the scenario.

For instance, you can add a neutral force of Terrans to the map and allow the player a chance to increase his force significantly by approaching them. You might set a timer, forcing the player to achieve the goal of destroying the enemy Zerg force quickly. Each possibility drastically changes the way the scenario plays, while keeping its original basic victory conditions.

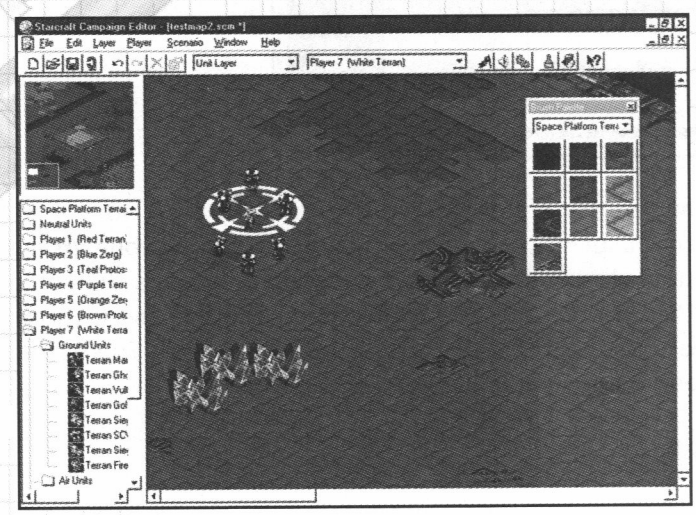


Fig. 7-4. Adding a rescuable force and a few triggers turns the simple scenario you created in Chapter 6 into a more complicated affair.

Or you can do something more complex. Consider the following:

- Add a neutral Terran force of Kerrigan and a half-dozen normal Marines (not the souped-up Marines you created in Chapter 4) to the lower-left portion of the map.
- Add a 30-minute countdown timer.
- Add the victory condition that Kerrigan must be rescued within 30 minutes (the extent of the timer).

And you can be even more inventive. Consider adding a trigger where, if the player reaches Kerrigan after 20 minutes, but before the timer runs out, Kerrigan becomes Infested Kerrigan—and joins the Zerg Force.

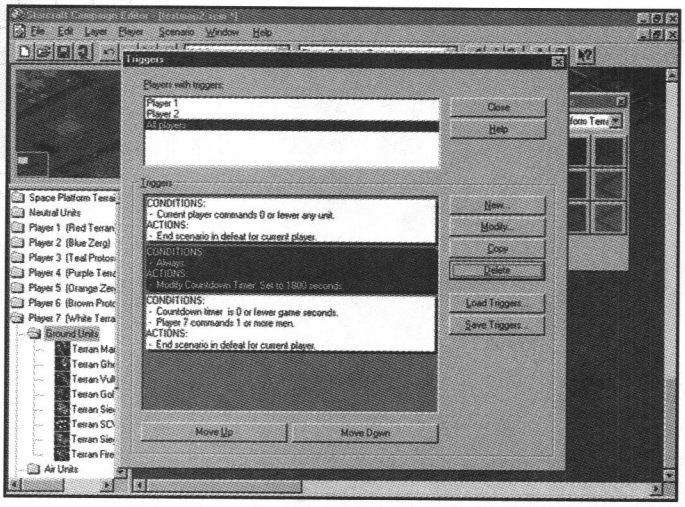


Fig. 7-5. Unless Kerrigan has been rescued, the player loses when the timer runs out.

Things have become really interesting, and there are multiple potential outcomes. If Kerrigan is reached quickly, the player adds a valuable character to his or her force. If the player gets there late, there's a significant enemy to deal with. If the player waits longer than 30 minutes, the player loses, because Kerrigan must be rescued before the timer runs out. You have two very different situations using the same scenario and the same victory conditions.

Your only limit is your imagination. Virtually anything is possible within the context of the *Starcraft* game. Zerg–Terran alliances, separating opposing forces with impassable terrain, even starting both sides with equal forces and no ability to gather resources.

Note

To create the foregoing scenario, your triggers would look something like this:

Condition

Countdown Timer is 600 game seconds or less (10 game minutes)

Player 1 (the Human Player) brings 1 of any unit to the Location where Kerrigan is

Player 1 has 2 or more opponents

Action

Remove Kerrigan and the Marine units

Create Infested Kerrigan and a half dozen Hydralisks at the Location

You could make this more palatable by adding a text message at 15 minutes left on the timer, having Kerrigan suggest that the Zerg are approaching her location.

Location Elevations

You can right-click on any location and select properties from the pop-up menu to edit location settings. Then check or uncheck the boxes to edit selected location.

For example, let's say your location encompasses a map section that includes both mid- and high-elevation terrain. However, you only want the location to apply to the high terrain. You can uncheck every box except for High Elevation. Units that walk on the middle elevation (or fly in the air above it) won't trigger the location.

Fig. 7-6. Controlling the computer AI gives you the chance to play with the complexity and difficulty of your scenario.

In fact, when you create a standard *Starcraft* mission, you should locate the mission's complexities here rather than at the victory condition level. Remember, you want to keep your players interested not only in the story line but also in their final goal. You can make a very complicated, difficult scenario with many possibilities and great replayability—and never change the default victory conditions.

Tertiary Triggers

This is where you give the scenario its "personality." Although you can produce a playable scenario without tertiary triggers, these enrich your scenario. They add the polish that makes a good scenario really memorable.

Just don't overdo it. As with any seasoning, adding the right amount will bring out the flavor in everything else. Too much can overwhelm the other ingredients: a single transmission, a .wav file or two, a text message, and maybe a score or leader board are all you really need to enhance the rest of your mission.

Single-Player vs. Multiplayer

When you develop your triggers, keep in mind whether your scenario will be played by a single human player or by several. The triggers you'll need will differ significantly—as well they should. Single-player and multiplayer *Starcraft* play far differently.

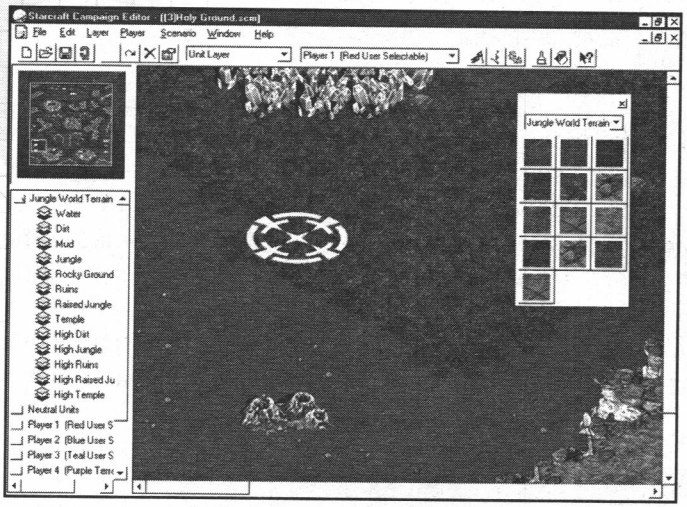


Fig. 7-7. Often, multiplayer scenarios such as this one will have only a basic set of triggers.

You'll want to keep most multiplayer games simple. Standard victory conditions of destroying all enemy structures will serve you well. You can throw in computer opponents, neutral forces, and rescuable armies for complexity. These can either hinder players who blunder into them or reward them for exploring the map efficiently. Give these armies random Start locations, and you can mix it up so players already familiar with the map will have no advantage over newer players.

Single-Player Triggers

Your single-player missions can be as elaborate as you like, as long as you either keep victory conditions simple or give them to the player in stages. A lone player battling a computer opponent or opponents has far more freedom than one fighting human foes: he or she can pause the game and think things through, for example, or save the game and come back to it later to try new strategies or explore different map areas.

As creator of the single-player scenario, you, too, can freely pause and unpause the game, using .wav files, text messages, and transmissions liberally to keep the player advised.

Unless they're critical to the mission's goal, usually you can ignore scores and leader boards in single-player. Keeping track of kills is unnecessary, unless the player's goal is to amass a given number of kills before the computer-controlled enemy does the same.

Multiplayer Triggers

Keep these short and to the point. You need assign only basic victory conditions to each player to create a successful multiplayer scenario. You may want to add some complexity: perhaps a computer opponent will attack the player with the most troops or the greatest number of kills at a certain point in the game. But for the most part, keep your basic multiplayer trigger set a little simpler; you have more players to create triggers for. It's too easy to get lost as you try to keep track of conditions for six potential players. By simplifying things, you make your job easier and are more likely to develop a memorable mission.

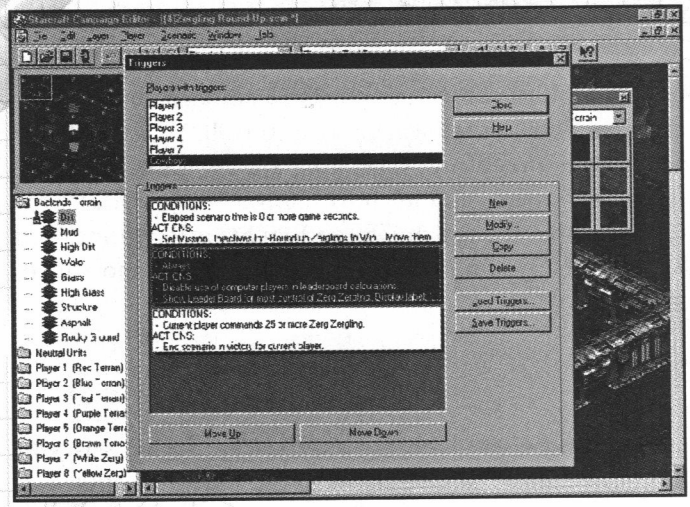


Fig. 7-8. Leader Board triggers, such as this one from the Zergling Round-Up scenario, work well in multiplayer games.

Unique Trigger Possibilities

Alarms

In multiplayer scenarios, consider including a warning sound when an enemy attacks. Setting up such an alarm system requires only a short buzzer or siren .wav file. The trigger would look something like this:

Condition—Non Allied Victory players (any enemy) bring 1 or more men to Location (player's base)
Action
    Play (filename).wav
    Wait 10000 (10 seconds)
    Preserve Trigger

Now when an enemy unit approaches the player's base, the .wav file will play and wait 10 seconds. If enemies remain in the player's base 10 seconds later, the alarm will sound again, and will go off every 10 seconds until the enemies are gone. (The alarm would have sounded only once without the "Preserve Trigger" action.)

Setting Traps

Fig. 7-9. Set the Doodad state to control such wall-mounted flamethrowers. They become active when the player approaches them.

Traps are unique to the Installation tileset. There are six trap types—floor gun and floor missile traps, right and left wall flame traps, and right and left wall missile traps. Putting a trap into operation requires using a Bring condition (usually) and a Set Doodad State action. The basic trap trigger looks like this:

Condition—Current Player brings 1 or more men to Location (of trap)
Action—Enable Doodad state for (name of trap Doodad) for (owner of trap) at Location

Doors work much the same way in an Installation mission. The condition remains the same, and the action is similar. The difference is that, to open a door, the door's Doodad state must be *disabled*, and that the door must be opened for all players.

Most doors operate only once in *Starcraft* missions. If you wish doors to close behind your players, you not only must preserve the open-door trigger, but you must add one to close the door when there are no units near it. This trigger would look like this:

Condition—Current player brings 0 or fewer men to Location
Action—Enable Doodad state for (door) for all players at Location

Map Revealing

Fig. 7-10. A Map Revealer allows you to show a player a particular part of the map at any time.

At times you may want to reveal a portion of the map to a player. This is fairly simple once you know the correct steps to take. Situations for revealing a previously hidden area might include one player taking a significant advantage in a multiplayer game, or the destruction of one force by another player. To create a Map Revealer, make the following trigger:

Conditions—(Any conditions you feel are appropriate for revealing a portion of the map)
Action
    Create Map Revealer (found in Neutral Units, Special in the menu) for Player (the player to see the portion of the map) at Location (the location to be revealed)
    Mini Map ping for current player at Location

You also can make the Map Revealer disappear after a given length of time, but this is a little more difficult. The Map Revealer isn't listed among the units available to be deleted. One way to do this is to set a switch (Switch 3, for example) in the foregoing trigger shown above. Add the following trigger:

Condition—Switch 3 (or any switch set by the reveal trigger) is set
Action
    Wait 20000 (20 seconds)
    Remove all units for Player (who received the Revealer) at Location

Taking Over Units and Structures

The basic *Starcraft* game includes no provision for players to take over enemy units and structures. By using triggers, though, you can have a Marine jump into an idle Siege Tank, or have a Hydralisk sneak into and claim a Protoss Forge.

Although it takes a little tinkering to make it work, it can be an intriguing change of pace for a player take over an enemy base rather than destroy it. Let's say you want Player 1's Ghost units to be able to take over Player 2's buildings. The trigger would look something like this:

Condition—Current Player brings 1 or more Ghosts to Location (of structure to be claimed)
Action
    Remove 1 Ghost for current Player 1 at Location
    Remove structure (the one taken over) for Player 2 at Location
    Create structure for Player 1 at Location

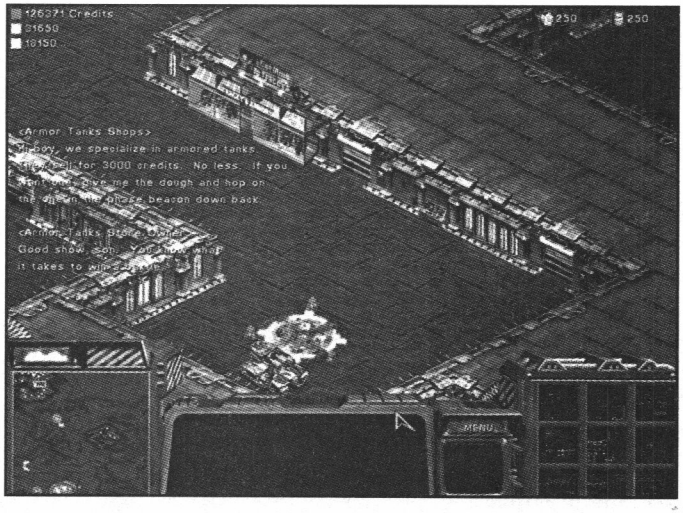


Fig. 7-11. A Terran civilian turns into a Siege Tank, thanks to this kind of trigger.

Using this sort of trigger requires many locations. Create the locations very tightly around the buildings, so Player 1 can't take over multiple structures simply by moving in a single Ghost unit. Also, this works best if the takeover unit is one that's difficult, time-consuming, or expensive to create, or one that's easily killed. The Terran Ghost is perfect, because the player must create a number of expensive structures to gain the ability to build them. The feeble Protoss Probe is another good choice. Or consider giving this ability to Zerg Queens.

Preventing Rushes

A basic *Starcraft* tactic, especially against other human players, is to amass a huge force of a relatively powerful units and simply swarm the enemy. Terran players can create a massed group of Goliaths, Zerg players can develop Hydralisks, and Protoss players can build a force of Dragoons, and use these to sweep over enemy bases quickly and efficiently. All these units can retaliate against both grounded and airborne enemies.

To prevent rushes early in your game, at a major cost to the offending player, use the following trigger:

Condition
- Non Allied Victory player (any enemy) brings 5 or more men to Location
- Elapsed time is 600 or fewer game seconds (10 minutes)

Action
- Kill all men for player at Location
- Display text message "Rushes are not allowed until 10 minutes have passed."
- Preserve Trigger

Note that once 10 game minutes have passed players can rush to their hearts' content. However, this stops a player from building up a large force and overwhelming another, unprepared player early in the game.

Preventing "Cheats"

One problem with developing a multiplayer scenario is that players find a number of ways to bend the rules. This is especially easy for an inventive player in a game where the number of kills determines victory. To prevent players from boosting their kills by slaughtering their own units, use the following trigger:

Condition
- Player 1 has suffered 10 deaths of any unit
- Player 2 kills score is 9 or fewer

Action—End scenario in defeat for Player 1

You can include a similar trigger for additional kill counts—perhaps every 10 or 25 kills. Also note that you'll run into some problems when players use units such as Siege Tanks, Reavers, and the Templars' Psychic Storm, because these units can and will damage allies as well as enemies.

Trading Resources Among Players

The basic *Starcraft* game includes no real way to allow players to trade resources, but you can. In fact, it's not even that difficult.

To create "trading posts," put down a beacon for each player at the center of the map, and create a location around each. Then set up the following trigger:

Condition—Player 1 brings 1 or more units to Location (of Player 2's beacon)
Action
    Modify resources for Player 1: subtract 100 ore and gas
    Modify resources for Player 2: add 100 ore and gas
    Preserve Trigger

Fig. 7-12. Stepping on the blue beacon transfers ore and gas from Player 1 to Player 2 in this mission.

Now, when Player 1 stands on Player 2's beacon, 100 units of ore and gas are subtracted from Player 1's total and added to Player 2's total. You also can do this with a switch, as follows:

Condition—Player 1 brings 0 or fewer units to Location (of Player 2's beacon)
Action—Clear Switch 1

Condition
    Player 1 brings 1 or more units to Location (of Player 2's beacon)
    Switch 1 is cleared
Action
    Modify resources for Player 1: subtract 100 ore and gas
    Modify resources for Player 2: add 100 ore and gas
    Set Switch 1

By stepping onto another player's beacon, the ore and gas transfer only once. To transfer more, the player (in this case Player 1) must leave the area to clear the switch, and then return.

Pitting Computer Opponents Against Each Other

There may be times when you want your computer forces not only to attack the human player, but to attack each other and add to the chaos. You can do this in several ways by running the appropriate AI script.

For instance, to have a computer-controlled Protoss force and a computer-controlled Terran force battle each other, you'd set up the triggers like this:

Condition (for the Protoss player)—Always
Action
    Run AI script (Protoss script) at Location (Protoss base)
    Run AI script Set player to Enemy at Location (Terran base)

Condition (for the Terran player)—Always
Action
    Run AI script (Terran script) at Location (Terran base)
    Run AI script Set player to Enemy at Location (Protoss base)

To have a computer enemy attempt to wipe out another computer enemy, include this trigger:

Condition—Always
Action—Run AI script Value this Location Higher at Location (enemy computer-controlled base)

This tells the AI to value the other base's location more highly, causing it to send units there and attempt to protect it from both human players and other computer opponents.

Two-Way Teleporters

Normally, allowing characters to teleport involves placing both a sending area and a receiving area, usually marked with a beacon, at each location. But you

also can create a way to have a single area at each location handle teleportation. To do this, set up a beacon at each location, and use the following trigger set:

Condition
    Player 1 brings 1 or more units to Location 1 (first teleport area)
    Switch 2 is cleared
Action
    Move units for Player 1 from Location 1 to Location 2 (second teleport area)
    Set Switch 1
    Preserve Trigger

Condition
    Player 1 brings 0 or fewer units to Location 1
    Switch 2 is set
Action
    Clear Switch 2
    Preserve Trigger

Condition
    Player 1 brings 1 or more units to Location 2
    Switch 1 is cleared
Action
    Move units for Player 1 at Location 2 to Location 1
    Set Switch 2
    Preserve Trigger

Condition
    Player 1 brings 0 or fewer units to Location 2
    Switch 1 is set
Action
    Clear Switch 1
    Preserve Trigger

When the player brings a unit to a teleporter, the unit is sent to the other location. The switches prevent the unit from instantly teleporting back. However, once the unit moves off the teleporter, the switches reset, allowing teleportation back to the original location.

Unusual Scenarios

There's no limit to what you can create using the StarEdit system, including scenarios with little or no similarity to a regular game of *Starcraft*. If you explore the scenarios included with *Starcraft*, you'll find a number of interesting possibilities, including a game of football, a destructive auto race, a simulation of herding creatures, and close approximations to classic arcade games of the past.

Fig. 7-13. The Race of Death scenario is just one unusual and interesting StarEdit possibility.

You can let your imagination run wild and create just about anything short of a flight simulator or first-person action game. Consider the following possibilities:

- Create a battle between players using only SCVs or Probes.
- Set up Capture-the-Flag games, where the players are penalized for killing opposing units.
- Create a maze and time players as they go from one end to the other.
- Have players run a gauntlet of enemies in an indestructible unit, with the shortest time or most kills winning.
- Create a death match, starting each player with a civilian; moving to particular locations allows them to gain powerful units or become indestructible briefly.
- Give the player a fully functional base and send waves of attacking enemies: as he or she defends successfully, the player receives extra units or defensive structures.
- Have SCVs and Wraiths as the only available units.
- Fill a map with damaged buildings: using a group of SCVs, the player must repair as many as possible before they're all destroyed, even as enemy units move in to attack the repaired buildings

All differ considerably from a normal game of *Starcraft*, but all are possible using StarEdit.

RPGs

One of the more interesting aspects of StarEdit is its potential for creating role-playing games. Here, the basic idea isn't necessarily to kill off enemy structures, but to complete a series of "quests" to achieve an ultimate goal. RPGs are complex and require far more effort to create. The result, however, is far more than a simple *Starcraft* scenario. Where even a lengthy, complicated *Starcraft* scenario might take a couple of hours, an RPG can take upward of 12 hours to play through.

The idea is a simple: rather than start with an army or base, the player starts with a single unit. This unit then must explore the surrounding landscape, killing enemies and, in doing so, becoming more powerful. Ultimately, the player and his or her character complete the task set before them.

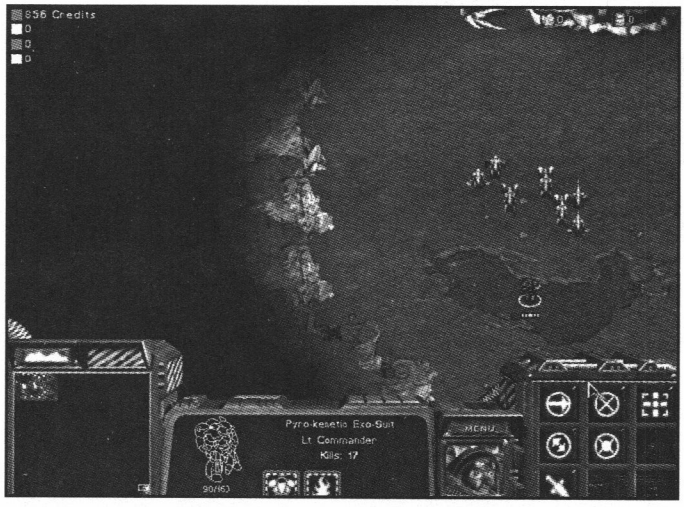


Fig. 7-14. A single unit versus a horde of enemies is stock in trade for an RPG. Make sure you boost the player's stats to give them a better chance of survival.

All RPGs share a few basic elements. Those you develop using StarEdit will contain these elements, as well.

First, there must be *progression*, wherein the character gains power and ability over time. Second, there should be an *ultimate goal* for the player's character to achieve. Third, *exploring* the surrounding area is critical.

Although it's easy to set up the second element using StarEdit, and the third resembles an Installation mission, there's no real way to increase power in a single unit or structure other than by upgrading. So you must go about incorporating that element differently.

By modifying the aforementioned "structure takeover" trigger set, you can simply allow the player to trade in his or her character for a more powerful unit.

Fig. 7-15. In this RPG, beacons allow a player with enough credits to convert a Terran civilian into a powerful unit.

Here are a few guidelines for developing an RPG using StarEdit:

- Use the largest map you can. This gives the player a huge area to explore, and makes the game itself bigger and more interesting.
- Prevent the player from building a base, either by limiting the resources the player can have at any one time or by vastly increasing the cost of creating structures.
- Boost Hit Point levels for the units the player will use. Because he or she will have a limited number of units, and because they'll be forced into combat repeatedly, making these units more powerful gives the player a better chance to survive.
- Create safe areas where the player can go for repair, or to get bigger and better units.
- Allow the player to gain control of additional units, thus creating a "party" of creatures. This allows the player to fight larger and tougher groups of enemies.

Fig. 7-16. Over time, you can allow the player to acquire more units—making an army, in effect, to fight the enemies located around the map.

- In addition to the ultimate goal, develop some intermediate objectives, such as rescuing characters or defeating certain armies, and reward the player with increased power or additional units.
- Intensify the challenge continually to keep the player interested.

Fig. 7-17. As enemy difficulty increases, make sure the player can increase his or her force.

- Never start the player with a frustratingly difficult task. Increase the scenario's difficulty gradually, as the player's power increases gradually. The mission should remain challenging, but never impossible.

A great way to control the increase in power is to limit the way the player gains new units or improves existing ones. Using a kill count, with upgrades available at given levels, simulates the progression of a normal role-playing game. Or give the player credits for every kill and the ability to use them for repairs, to trade up to better units, or to purchase additional units.

Because an RPG scenario differs completely from a normal *Starcraft* mission, it's necessarily trigger- and location-heavy. You'll need triggers to control the players' ability to purchase additional units and upgrade those they have, as well as to control enemy actions.

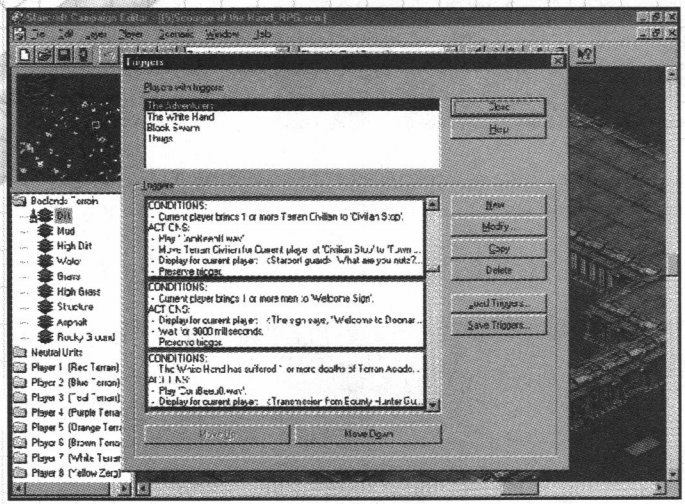


Fig. 7-18. The trigger list for an RPG such as this one can run into the hundreds. Take it slow and don't let it overwhelm you.

The most important rule in developing an RPG is to *save and playtest constantly*. Make sure your triggers work the way they're supposed to. After you've playtested the game and are satisfied with the scenario, playtest again. The players who go through your scenario will think of things you won't, and may find errors you overlooked. Get as much feedback as possible, and keep refining your scenario.

Chapter 8 StarEdit Extras

The way you'll use the AI in *Starcraft* will vary considerably from scenario to scenario. Chapter 6 touched on setting the AI for computer-controlled forces. This chapter examines AI function in depth. Learning how it operates will give you a better framework for determining how to use the AI in your own scenarios.

Custom AI

- These AIs expect to start with nothing, or very little.
- Custom, or Melee, AIs expand when their resources run low or when their script instructs them to, whichever comes first.
- They build a lot of SCVs, Drones, or Probes, expecting to start with only 50 units of minerals.
- They use scripted rushes, but only when they have a good chance to end the scenario. They're unlikely to use these rushes if the human player starts with any units.
- They attack when a human player starts a new base, in an effort to box the player in. If the player doesn't expand within 40 game minutes or so, they'll attack the player's main base.
- They respond aggressively to all player attacks.

Campaign AIs

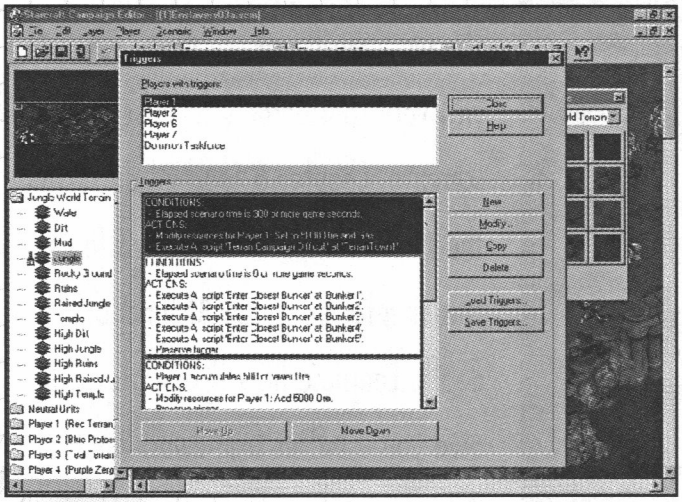


Fig. 8-1. Campaign AIs are designed to be beaten, but more difficult AIs, such as the one used in this scenario, offer a greater challenge.

- These AIs expect to begin with a town, and so won't build new structures. They'll repair or replace important structures that are damaged or destroyed.
- They won't expand to new bases.
- They expect to start with numerous resources, and will build few gathering units.
- They use no rushes, and won't attempt to destroy the player.
- They'll attack what they find, and won't specifically target the player if he or she tries to expand to a new base.
- They attack the player's bases periodically, regardless of the composition of the player's army.
- They don't respond aggressively to attacks, often giving the player time to destroy a structure or two.

Choosing the Correct AI

Choosing the correct AI is crucial for a successful scenario. For the greatest challenge, you'll want to pit the human player against the Custom AI, but this isn't the best choice if:

the player starts with a larger force than the computer AI;
the player doesn't have a base to attack;
the player doesn't need to expand for additional resources;
the scenario's triggers complicate the mission beyond just fighting.

Campaign AI settings work better in these cases. For Installation missions, it's best to set the AI for all computer-controlled units to "Send All Units on Random Suicide Missions," so they'll attack the human player's troops whenever they get close.

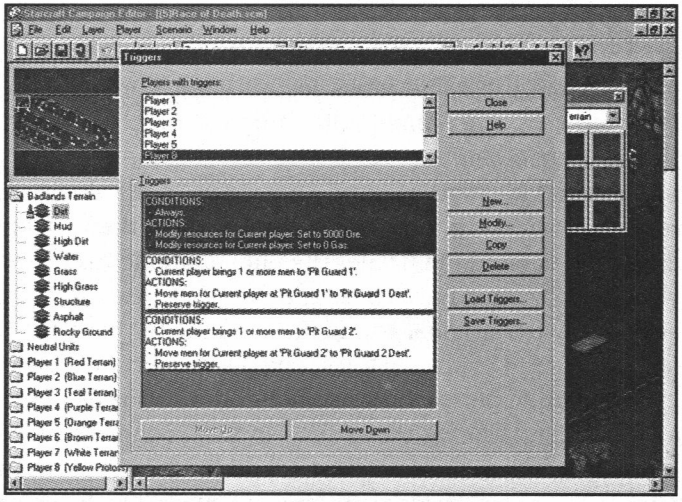


Fig. 8-2. If you don't set an AI for a computer force, it will only defend, and won't harvest ore or gas.

Tip

If you're creating a campaign, stick with a Campaign AI. It gives the human player the best chance of getting through all the scenarios.

Playing with Sound

At times you may want to change the sound effects used with the game's units and structures, and/or to use some of the game's own sound files. To extract the sound files, you'll first need to download the *SCsounds* utility from Blizzard's Web site at http://www.battle.net/faqs/starcraft/features.htm. You'll need to unzip it, and you must have 80 MB of available hard drive space

to store the sound files. Once you extract the files, you can use *Starcraft*'s original sounds in your own scenarios.

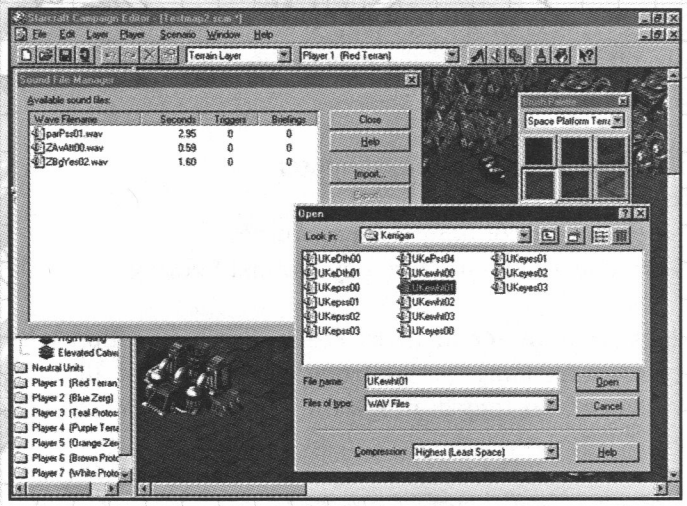


Fig. 8-3. The *SCsounds* utility gives you access to all the unit and structure sounds in *Starcraft*.

It's also possible to replace the original unit sound with another sound. Each *Starcraft* sound has a name the game looks for on your *Starcraft* CD—unless you have a Sounds folder in your *Starcraft* directory on your hard drive. The *SCsounds* utility creates this folder.

To exchange sound files, first locate the sound you wish to alter. Delete it, and replace it with the desired sound in the same directory. Give the new file the same name as the deleted file. To get the old sound file back, just delete the new file from your hard drive directory: if *Starcraft* can't find it there, it will play the original sound file from the CD by default.

Let's say you wanted to change the sound for the Terran advisor, for when the player attempts to build something but doesn't have enough Vespene gas. First, locate that sound in the sound directory under Sound\Terran\Advisor; the particular sound file is taderr01.wav. Then delete it, name your new file taderr01.wav, and place it in the same directory. Now, when the Terran player attempts to build something and doesn't have enough gas, *your* sound file will play. If you want to revert to all of the original game sounds, simply delete the Sounds folder from your hard drive.

Tip

In the Terran folder, you'll find sounds for the Science Vessel in the Vessel folder, and for the Wraith in the Phoenix folder. Sounds for the Protoss Reavers are in the Trilobyte folder, and the Observers' sounds are in the Witness folder. You'll find the Zerg Scourges' sounds in the Avenger folder.

Creating Random Events

A limitation of the trigger system is that all events are planned from the beginning. That is, a particular event will happen at a certain time or at a certain location when specific conditions are met. However, there is a way to create a kind of randomness, using a combination of switches and triggers.

Each switch can be either set or cleared. Hence, a trigger that uses a switch can manifest two outcomes—what happens when the switch is set and what happens when the switch is cleared. A trigger that depends on two switches can generate four outcomes—both switches set, both cleared, the first switch set and the second cleared, and the first switch cleared and the second switch set. A trigger using three switches can generate eight possible outcomes, and so on.

Consider the possibilities using only three switches. These examples use your scenario from earlier. Player 1 is your human player, Player 2 your Zerg computer-controlled enemy. You'll create a trigger like this:

Condition—Always
Action
    Toggle Switch 1
    Toggle Switch 2
    Toggle Switch 3
    Preserve Trigger

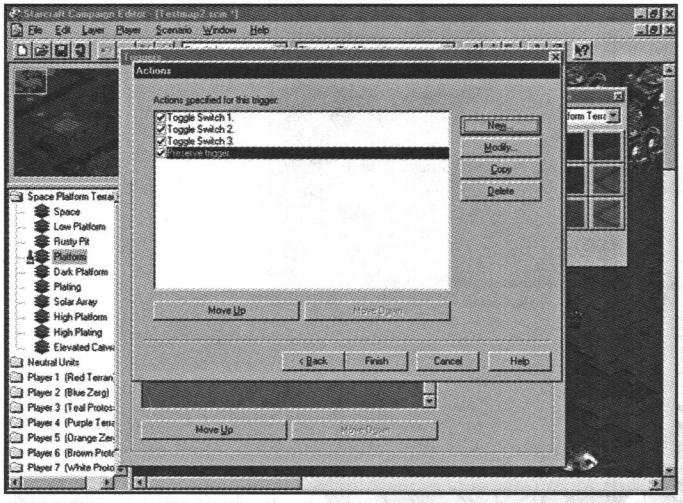


Fig. 8-4. These switches will toggle every few seconds.

Now, every few seconds, all three Switches will toggle, either going from cleared to set or from set to cleared. This doesn't insert any randomness yet, however. To do that, you must add some additional triggers to toggle the switches. For example:

Condition—Player 1 brings 1 or more SCV to Location 1 (the central platform)
Action
    Toggle Switch 1
    Preserve Trigger

Condition—Player 1 accumulates 1000 or more ore
Action
    Toggle Switch 2
    Preserve Trigger

Condition—Player 1 commands 70 or more men
Action
    Toggle Switch 3
    Preserve Trigger

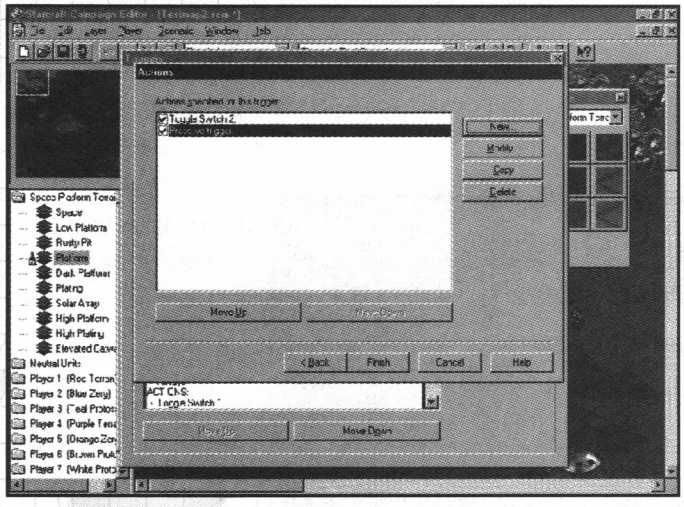


Fig. 8-5. Now other game events can affect the switches. This allows them to go out of sequence, making random events possible.

Using these three triggers, you can knock your switches out of sequence. Now you can create some triggers based on the potential permutations of the switch positions. For example:

Condition

Switch 1 is set
Switch 2 is set
Switch 3 is cleared

Action—Create Ultralisk at Location 1 (the central platform) for Player 2

Condition

Switch 1 is cleared
Switch 2 is set
Switch 3 is cleared

Action

Kill all Terran Marine units for Player 1 at Location 1
Display text message for Player 1: "Sir, the Zerg appear to have developed some sort of virus. All of our Marine units at the second base have been killed!"

Condition

Switch 1 is cleared
Switch 2 is set
Switch 3 is set

Action

Create Terran Battlecruiser at Location 0 (player's base) for Player 1
Display text message for Player 1: "Thought you could use a little extra help, sir."

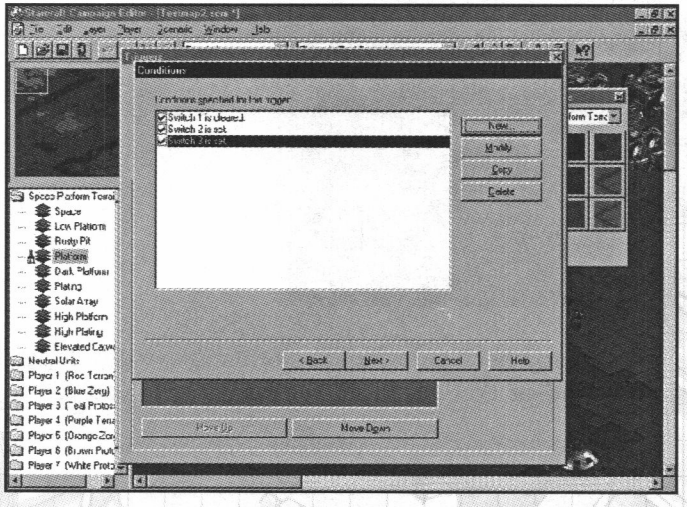


Fig. 8-6. With a little luck, the player will score an extra Battlecruiser to use in the scenario.

Events triggered by the foregoing sequence may or may not occur during the course of the game, and each will happen only once. In fact, there's no guarantee any of them will happen. But because you've toggled the switches back and forth and have added various ways for the player to toggle them, you've created the *possibility* these events may occur. A player conceivably could play this scenario several times and experience different events each time.

Three switches gives you eight possible combinations for generating random events. The maximum number of 32 switches gives you 4,294,967,296 possible combinations—more than you ever could use.

What To Do Next

Now that you understand the basics of creating your own *Starcraft* scenario, what should you do next? The easy answer is *whatever you want*, but I'll make a few suggestions.

Make sure you check out Blizzard's Web site, http://www.blizzard.com, for updates on *Starcraft* and the StarEdit engine. Plans are in the works to add things to *Starcraft*, including new heroes for each species, the ability to label triggers and switches, and increasing the number of switches and locations available. StarEdit is more than just a tool for creating your own scenarios. It's a dynamic process, and Blizzard will update and change it continually.

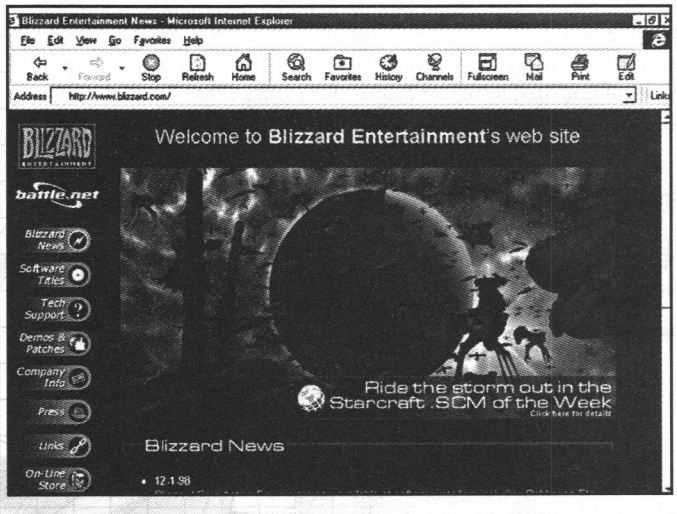


Fig. 8-7. Check Blizzard's Web site for updates on *Starcraft* and StarEdit.

Finally, explore the Internet to check out scenarios others have developed. They're sure to have tricks and ideas that may not have occurred to you yet.

Appendix A: Keyboard Shortcuts

File Menu Alt + F

KEYBOARD COMMAND	FUNCTION
Ctrl + N	New Map
Ctrl + O	Open Map
Ctrl + S	Save Current Map
Ctrl + C	Compact Map

Edit Menu Alt + E

KEYBOARD COMMAND	FUNCTION
Ctrl + Z	Undo
Ctrl + Y	Redo
Delete	Clear
Ctrl + A	Select All
Alt + G	Show/Hide Grid
Enter	Properties Menu (when unit or structure is clicked on)

Layer Menu Alt + L

KEYBOARD COMMAND	FUNCTION
Ctrl + T	Terrain Layer
Ctrl + D	Doodad Layer
Ctrl + U	Unit Layer
Ctrl + L	Location Layer
Ctrl + F	Fog of War Layer

Player Menu Alt + P

Keyboard Command	Function
Ctrl + G	Go to Start Location for current player
Ctrl + Enter	Setting Menu

Scenario Menu Alt + S

Keyboard Command	Function
Ctrl + Shift + F	Forces Menu
Ctrl + Shift + S	Sounds Menu
Ctrl + Shift + T	Triggers Menu
Ctrl + Shift + B	Mission Briefings Menu
Ctrl + Shift + H	Unit and Heroes Settings Menu
Ctrl + Shift + U	Upgrade Settings Menu
Ctrl + Shift + A	Special Abilities Settings Menu
Ctrl + Shift + Enter	Properties Menu

Window Menu Alt + W

Keyboard Command	Function
Ctrl + B	New Brush Palette
Ctrl + P	New Doodad Palette
Ctrl + W	Open New Map View

Help Menu Alt + H

Keyboard Command	Function
F1	Help Menu

Appendix B: Unit Data

Terran Units

Unit Type	Armor Type	Supply Cost	Gas Cost	Mineral Cost	Hit Points	Ground Attack Strength	Air Attack Strength	Reload Rate
Battlecruiser	Heavy	8	300	400	500	25*	25*	50
Dropship	Heavy	2	100	100	150	0	0	0
Firebat	Light	1	25	50	50	16†‡	0	37
Ghost	Light	1	75	25	45	10†	10†	37
Goliath	Heavy	50	100	125	10	20	37	
Marine	Light	1	0	50	40	6	6	25
Missile Turret	Heavy	0	0	100	200	0	20*	25
Science Vessel	Heavy	2	300	25	200	0	0	0
SCV Light	1	0	50	60	5	0	25	
Siege Tank, Siege Mode	Heavy	2	100	150	150	70*‡	0	125
Siege Tank, Tank Mode	Heavy	2	100	150	150	30*	0	62
Vulture	Medium	2	0	75	80	20†	0	50
Wraith	Heavy	2	100	200	120	8	15*	37

* Explosion attack does 50% damage to light armor, 75% damage to heavy armor.
† Attack does 25% damage to heavy armor, 50% damage to medium armor.
‡ Attack damages nearby units and structures as well as the main target.

Zerg Units

Unit Type	Armor Type	Supply Cost	Gas Cost	Mineral Cost	Hit Points	Ground Attack Strength	Air Attack Strength	Reload Rate
Broodling	Light	1	0	0	30	4	0	25
Defiler	Medium	2	100	25	80	0	0	0
Drone	Light	1	0	50	40	5	0	37
Egg	Heavy	0	0	1	200	0	0	0
Guardian	Heavy	2	100	50	150	20	0	50
Hydralisk	Medium	1	25	75	80	10*	10*	25
Infested Terran	Light	1	50	100	60	500	0	50
Larva	Light	0	0	1	25	0	0	0
Mutalisk	Light	2	100	100	120	9	9	37
Overlord	Heavy	-7	0	100	200	0	0	0
Queen	Medium	2	150	100	120	0	0	0
Scourge	Light	.5	75	25	20	0	110	50
Sport Colony	Heavy	0	0	50	400	0	15*	25
Sunken Colony	Heavy	0	0	75	400	30*	0	37
Ultralisk	Heavy	6	200	200	400	20	0	25
Zergling	Light	.5	0	50	35	5	0	15

* Explosion attack does 50% damage to light armor, 75% damage to heavy armor.

Protoss Units

Unit Type	Armor Type	Supply Cost	Gas Cost	Mineral Cost	Shield Strength	Hit Points	Ground Attack Strength	Air Attack Strength	Reload Rate
Arbiter	Heavy	4	500	25	150	200	10*	10*	75
Archon	Heavy	4	300	100	350	10	30	30	25
Carrier	Heavy	8	300	350	150	250	5	5	4
Dragoon	Heavy	2	50	150	80	100	20*	20*	50
High Templar	Light	2	150	50	40	40	0	0	50
Observer	Light	1	75	25	20	40	0	0	0
Photon Cannon	Heavy	0	0	150	100	100	20*	20*	37
Probe	Light	1	0	50	20	20	5	0	37
Reaver	Heavy	4	100	200	80	100	100*‡	0	50
Scout	Heavy	2	100	300	90	130	8	24*	37
Shuttle	Heavy	2	0	200	60	80	0	0	0
Zealot	Light	2	0	100	80	80	16	0	37

* Explosion attack does 50% damage to light armor, 75% damage to heavy armor.
‡ Attack damages nearby units and structures as well as the main target.

Structure Data

Terran Structures

Structure	Armor	Gas Cost	Mineral Cost	Hit Points	Build Time
Command Center	Heavy	0	400	1,500	120
Supply Depot	Heavy	0	100	500	40
Refinery	Heavy	0	100	750	40
Barracks	Heavy	0	150	1,000	80
Engineering Bay	Heavy	0	125	850	60
Academy	Heavy	0	200	600	80
Bunker	Heavy	0	100	350	30
Factory	Heavy	100	200	1,250	80
Armory	Heavy	50	100	750	80
Starport	Heavy	100	200	1,300	80
Science Facility	Heavy	200	150	850	80
Machine Shop	Heavy	50	50	750	40
Control Tower	Heavy	50	100	500	40
Physics Lab	Heavy	50	50	600	40
Covert Operations	Heavy	50	50	750	40
Nuclear Silo	Heavy	100	100	600	80
Comsat Station	Heavy	50	50	500	40

Zerg Structures

Structure	Armor	Gas Cost	Mineral Cost	Hit Points	Build Time
Hatchery	Heavy	0	300	1,250	100
Creep Colony	Heavy	0	75	400	20
Extractor	Heavy	0	50	750	40
Spawning Pool	Heavy	0	150	750	80
Evolution Chamber	Heavy	0	75	750	40
Hydralisk Den	Heavy	50	100	850	40
Spire	Heavy	150	200	600	120
Queen's Nest	Heavy	100	150	250	40
Nydus Canal	Heavy	0	150	250	40
Ultralisk Cavern	Heavy	200	150	600	80
Defiler Mound	Heavy	100	100	850	60
Greater Spire	Heavy	150	100	1,000	100
Lair	Heavy	100	150	1,800	100
Hive	Heavy	150	200	2,500	120

Protoss Structures

Structure	Armor	Gas Cost	Mineral Cost	Shield Strength	Hit Points	Build Time
Gateway	Heavy	0	150	500	500	60
Forge	Heavy	0	200	550	550	40
Cybernetics	Heavy	0	200	500	500	60
Shield Battery	Heavy	0	100	200	200	30
Robotics Facility	Heavy	200	200	500	500	60
Stargate	Heavy	200	200	600	60	80
Citadel of Adun	Heavy	100	200	450	450	60
Robotics Support Bay	Heavy	100	50	450	450	30
Observatory	Heavy	100	150	250	250	60
Fleet Beacon	Heavy	200	300	500	500	60
Templar Archives	Heavy	200	100	500	500	60
Arbiter Tribunal	Heavy	150	200	500	500	60
Nexus	Heavy	0	400	750	750	120
Pylon	Light	0	100	300	300	30
Assimilator	Heavy	0	100	450	450	40

Index

A
ability settings. *See* special abilities.
Accumulate condition, 91-92
action. *See* also specific actions.
    adding to mission briefing, 74
    defined, 5, 90, 97
Actions menu, 74
Addons folder, 25, 62
AI scripts
actions, 99-100
    choosing correct, 135
    Custom and Campaign, 134
    Starcraft's contrasted with StarEdit's, 6-7
    tips for using, 100, 126, 135
air attack strength values, 143-144
Air Units folder, 25
alarm trigger, 120-121
Allied Victory setting, 20
Allies setting, 20
Always conditions, 92
animations, 77, 103
arcade games, 128
armies
    altering species of, 58-59
    organizing, 20-21
armor, 67, 143-145
art, Starcraft, 7
Artificial Intelligence. *See* AI scripts.
asymmetrical maps, 53-55
attacks
    AI script for, 126
    unit data, 143-145
auto race scenario, 128

B
base defenses, 42-46
Blizzard Entertainment, 36, 135, 140
bridges, 27, 45. *See also* Doodads.
briefing. See mission briefing.
Bring conditions, 93
Brush Palette, 20, 23, 26-27, 35
build time, structure, 144-145
Buildings folder, 25
Burrowed setting, 18

C
campaign
    AI scripts, 100, 134
    defined, 3
    linking missions/scenarios into, 4, 71, 86-88, 98
Capture-the-Flag games, 128
Carrier Capacity upgrade, 18
Cascade option, 35
Center View action, 101, 107
cheats, preventing, 124
cinematics, 7
Clear command, 17, 29
cliffs, 27
Cloaked setting, 18
Cloaking ability, 69
Close Map View options, 35
color code, Mini Map, 24
combat missions, 12
Command conditions, 93-94
Command Palette, 3
Compact button, 16, 28
computer-controlled units, 9, 30, 60, 107
condition
    defined, 5, 90
    setting, 87
    specific types
        Always/Never, 92
        Bring, 93
        Command, 93-94
        Kills, 95-96
        Resource, 91-92
        Score, 96
        Time, 94-95
    variables, 91
Control actions, Leader Board, 105-106
control options, unit, 60-61
Control Picker, 30
costs
    special abilities, 33-34, 69
    supplies and resources, 143-145
    upgrade, 68
Countdown Timer action/condition, 94, 108
Covert Ops, 62
Create Unit actions, 102
Critters, 24, 25
current player, 90
Custom AI, 100, 134
custom missions, 12-13

D
Dark Platform, 49
Deaths condition, 95
Defeat action, 98, 114
defenses, base, 42-46
Delete key, 17
difficulty-level, controlling scenario's, 55, 61, 62, 131
Diplomatic Settings menu, 21
Display Text Message action, 102
Doodad Palette, 23, 27, 35, 44
Doodads
    defined, 19, 45
    placing, 27-28, 44
    setting states of, 104
doors, 45, 104, 121. *See also* Doodads.
Drones, 42
duration, trigger, 90

E
Edit menu, 28-29, 141
editing
    maps, 3, 18-19, 24
    scenarios, 4
Elapsed Time condition, 95
Energy Points, 18
"Enter Nearest Bunker" AI, 100
experience points, 13
explosion attacks, 143-144

F
File menu, 28, 141
files
    saving, 16, 28, 132
    sound, 21, 79-80, 102-103, 135-136
    squashing, 16
    StarEdit Help, 23, 36, 38
Firebats, 62
Flags, 25
floor traps, 121
Fog of War, 19
folders, 24-26, 62
football scenario, 13, 128
forces, dividing into opposing sides, 66
Forces button, 20-21, 66
Forces menu, 30
Free for All mode, 8-9

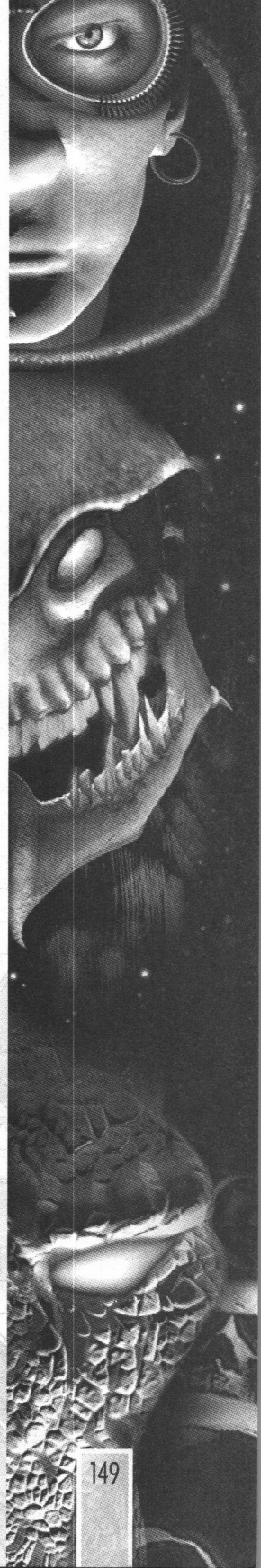
G
Game Type options, 8-9
gas. *See* resources.
Ghost units, 62
goals
    Leader Board actions, 105, 106-107
    for RPG scenarios, 129, 130
    scenario/mission, 72
graphics, *Starcraft,* 7
Greed action, Leader Board, 106
grid, map, 29
ground attack strength values, 143-144
Ground Units folder, 25
gun traps, 121
guns, activating, 104-105

H
Hallucinated setting, 18
health of unit. *See* Hit Points.
Help button, 23
Help menu, 36, 38, 142
Hero settings. *See* Unit and Hero Settings menu.
Heroes folder, 25
Hide Portrait action, 77-78
High Platform, 47-48
High Plating, 49-50
Highest Score condition, 96
Hit Points, 18, 67, 130, 143-145
human-controlled units, 30, 60

I
impassable terrain, 25, 53
In Hangar setting, 18
In Transit setting, 18
interface, StarEdit, 15
Internet, 140
Invincible setting, 18, 105

K
keyboard shortcuts, 36, 141-142
Kill Unit actions, 104
Kills actions, Leader Board, 106, 107
Kills conditions, 95-96

L
Layer menu, 29, 141
Layer Picker, 18-19
layers, map, 18-19



Leader Board actions, 105-107
Least Kills condition, 96
Least Resources condition, 92
licensing agreement, StarEdit, 36
linking missions/scenarios, 4, 71, 86-88, 98
Location layer, 90
Low Platform, 42-44
Lowest Score condition, 96

M
Main Toolbar, 16
map editing, 3, 18
Map Revealers, 25, 122
maps. *See also* melee map.
    asymmetrical, 53-55
    creating, 16, 38-42
    editing, 18-19, 24
    opening, 16
    playtesting, 41, 51
    resource distribution, 53
    safe areas, 130
    saving, 16
    size considerations, 16, 51
    switching between, 36
    viewing, 35, 122
Marines, 62
mazes, 128
melee map
    base defenses, 42-46
    central resource deposit, 46-48
    Doodads, 50
    minimum requirements for, 42
    playtesting, 51
    simplicity of, 11-12
    steps for creating, 38-42
    terrain types, 49
Melee mode, 8-9
menus, pull-down, 28-36, 141-142
millisecond, 77
Mineral Fields, 18, 25, 48
minerals. *See* resources.
Mini Map
    color code, 24
    Ping action, 101
    purpose of, 24
missile traps, 121
Mission actions, 97-98
mission briefing

creating, 73-80
deleting, 83
pausing, 83, 84
purpose of, 4-5, 31
samples, 80-86
Mission Briefings menu, 31
Mission Objectives action, 75-76, 97
missions. *See also* scenarios.
common types, 11-12
defined, 3
linking, 4, 71, 86-88, 98
major parts of, 3-6
playing, 8
standard vs. custom, 12
moats, 42-43
Most Kills condition, 96
Most Resources condition, 92
Move Location action, 101
Move Unit action, 101-102
multiplayer games, 8, 11, 118-119, 120
Mute Unit Speech action, 103

N
neutral units, 24, 25, 30, 60, 61, 107
Never conditions, 92
New Brush Palette button, 23, 26
New button, 16, 28
New Doodad Palette button, 23
New Map View option, 35
non-unit objects. *See* Doodads.
nuclear silos, 62

O
objective, mission, 71, 75-76, 97
obstacles. *See* Doodads.
Open button, 16, 28
Opponents condition, 96

P
palettes
Brush, 20, 23, 26-27, 35
Command, 3
Doodad, 23, 27, 35, 44
Pause actions, 107
Ping action, 101
pits, 53
Play Custom option, 8
Play Wav action, 79-80, 103
Player Grid, 30
Player menu, 29-30, 142

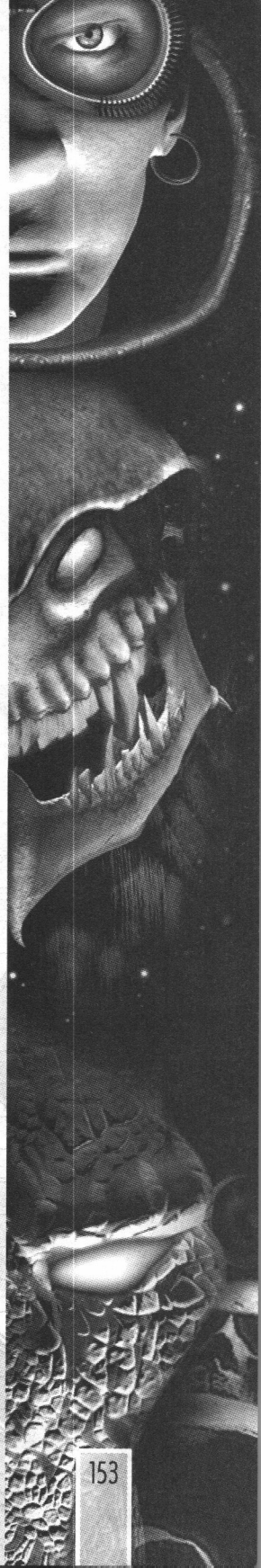
Player Picker, 20, 26
Player Settings, 20, 30, 58-65
players. *See also* units.
    allocating resources to, 98
    color code, 24
    creating mission briefings for, 74
    current, 90
    organizing within scenarios, 20-21, 66
playtesting, 41, 51, 70, 132
Points actions, Leader Board, 106, 107
portraits, 76-78, 79, 82, 103
power settings, 66
Power Ups folder, 26
pre-mission briefing. *See* mission briefing.
Pro Bowl mission, 13
Probes, 42
progression, RPG, 129
Properties button, 17-18
Properties menu, 17-18, 29, 34
Protoss units and structures, 59, 67, 144, 145
pull-down menus, 28-36, 141-142

Q
quests, 129

R
race car scenario, 128
Race of Death scenario, 128
Race Picker, 30
ramps, 27, 44, 45. *See also* Doodads.
random events, 137-140
Random Start Location setting, 20
Reaver upgrade, 18
Redo command, 17, 28
reload rate, 143-144
Remove Unit actions, 104
rescuable units, 30, 60, 61, 99, 107
Resource conditions, 91-92
resources
    allocating, 53, 98
    asymmetrical, 54-55
    changing properties of, 17, 18
    color code, 24
    conditions, 91-92
    costs, 143-145
    creating central deposit on melee map, 46-48
    painting over, 49
    selecting, 17
    trading, 125-126
Resources actions, Leader Board, 106, 107

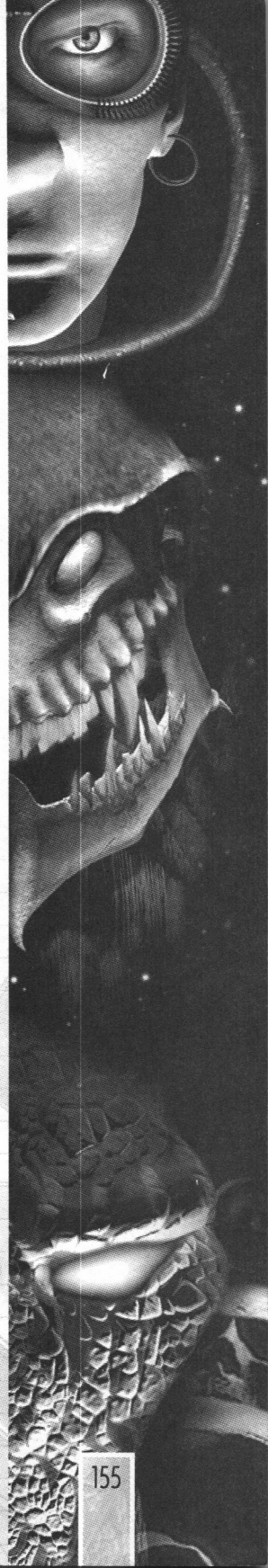
role-playing games. *See* RPG scenarios.
RPG scenarios
    basic elements of, 129
    challenges to creating, 13, 129
    contrasted with normal *Starcraft* missions, 132
    guidelines for creating, 130-132
    triggers for, 132
Run AI Script actions, 99-100
rushes, preventing, 124
Rusty Pits, 50

S
safe areas, 130
Save As button, 28
Save button, 16, 28
Scenario menu, 30-34, 142
scenarios. *See also* missions.
    controlling difficulty level, 55, 61, 62, 131
    customized, 10
    defined, 3
    designing, 2
    editing, 4
    naming, 34-35
    organizing players within, 20-21, 66
    playtesting, 70, 132
    setting goals for, 72
    setting victory conditions, 114-118
    single- vs. multiplayer, 118-120
    suggestions for unusual, 128-129
    transferring settings between, 88
score, setting, 98
Score conditions, 96
scripts. *See* AI scripts.
SCsounds utility, 135-136
SCVs, 42
secondary triggers, 115-118
Select All command, 29
Set Doodad State action, 104-105
Set Invincibility action, 105
Set Mission Objectives action, 97
Set Next Scenario action, 98
Set Resources action, 98
Set Score action, 98
Setting Countdown Timers action, 108
shape asymmetry, 54
Shared Vision setting, 20
shield strength values, 18, 144, 145
shields, 67
shortcuts. *See* keyboard shortcuts.
Show Grid command, 29



Show Portrait action, 76, 79
Sidebar, 24
Siege Tanks, 45, 62
silos, nuclear, 62
single-player games, 8, 118-119
Sound actions, 102-103
Sound File Manager, 21-22
sound files, 21, 79-80, 102-103, 135-136
Sounds button, 21-22
Sounds menu, 30
speaking portraits, 77, 79, 82, 83-85, 103
special abilities, 33-34, 64-65, 69
Special Buildings folder, 26
Special folder, 25
species options, 58-59
squashing map files, 16
stairways, 27, 45. *See also* Doodads.
Starcraft
    Campaign Editor (See StarEdit)
    licensing agreement, 36
    updates, 140
StarEdit
    Help files, 23, 36, 38
    interface, 15
    launching, 38
    licensing agreement, 36
    major functions, 3-6
    purpose of, 2
    strengths and weaknesses, 6-8
    updates, 140
Start locations
    folder, 26
    placement of, 53
    random, 20
strength values, 143-145
Structure Data tables, 144-145
structures. *See also* units.
    changing properties of, 17
    controlling, 104-105
    data tables, 144-145
    destroying, 104, 123
    folder, 25
    rescuing, 61
    selecting, 17
    taking over, 123
subfolders, player, 25-26
suicide missions, 99
switch, trigger, 90
switches, 90, 97, 100-101, 137-140, 140

T
Talking Portrait action, 103
teleporters, 126-127
terrain
    characteristics, 25
    choosing, 39
    folder, 24-25
    impact on game-playing, 25
Terran Command Center, 58
Terran structures, 144
Terran units, 59, 62, 143
tertiary triggers, 118
testing scenarios, 41, 51, 70, 132
Text actions, 102
Text Message action, 78-79, 102
theme, mission, 71, 86
Tile options, 35
tilesets, 38, 39
Time conditions, 94-95
timed commands, 80
timer, countdown, 94, 108
tips
    accessing StarEdit's Help function, 38
    defining map locations, 93
    disabling unit types, 62
    placing building for Vespene gas, 73
    placing Doodads, 27, 28
    placing units, 26
    removing units and resources from map, 41
    selecting units, 17
    upgrading Carrier and Reaver capacity, 18
    using layers to edit map parts, 18
    using New Scenario dialog box, 39
    where to find sound files, 21, 136
    working with folders, 25
Tool Tree, 24, 26
toolbars, 16
trading posts, 125-126
Transmission action, 79, 103, 107
traps, 45, 121. *See also* Doodads.
trenches, 45-46
trigger actions
    defined, 97
    preserving, 108
    specific types
    AI, 99-100
    Create, 102
    Leader Board, 105-107
    Mission, 97-98



Pause, 107
Preserve Trigger, 108
Setting Countdown Timers, 108
Setting Switches, 100
Text and Sound, 102-103
Unit, 104-105
Victory/Defeat, 98-99
View, 101
triggers
adding and removing, 5
built-in, 5
for creating random events, 137-140
defined, 5, 89, 90
how they work, 90
labeling, 140
linking missions with, 86-88
for RPG scenarios, 132
sample, 108-111, 120-127
setting conditions for, 87
single- vs. multiplayer, 118-120
unique, 120-127
for Victory and Defeat conditions, 114-118
Triggers button, 22-23
Triggers menu, 31
turning point, 93. *See also* victory.
two-way teleporters, 126-127

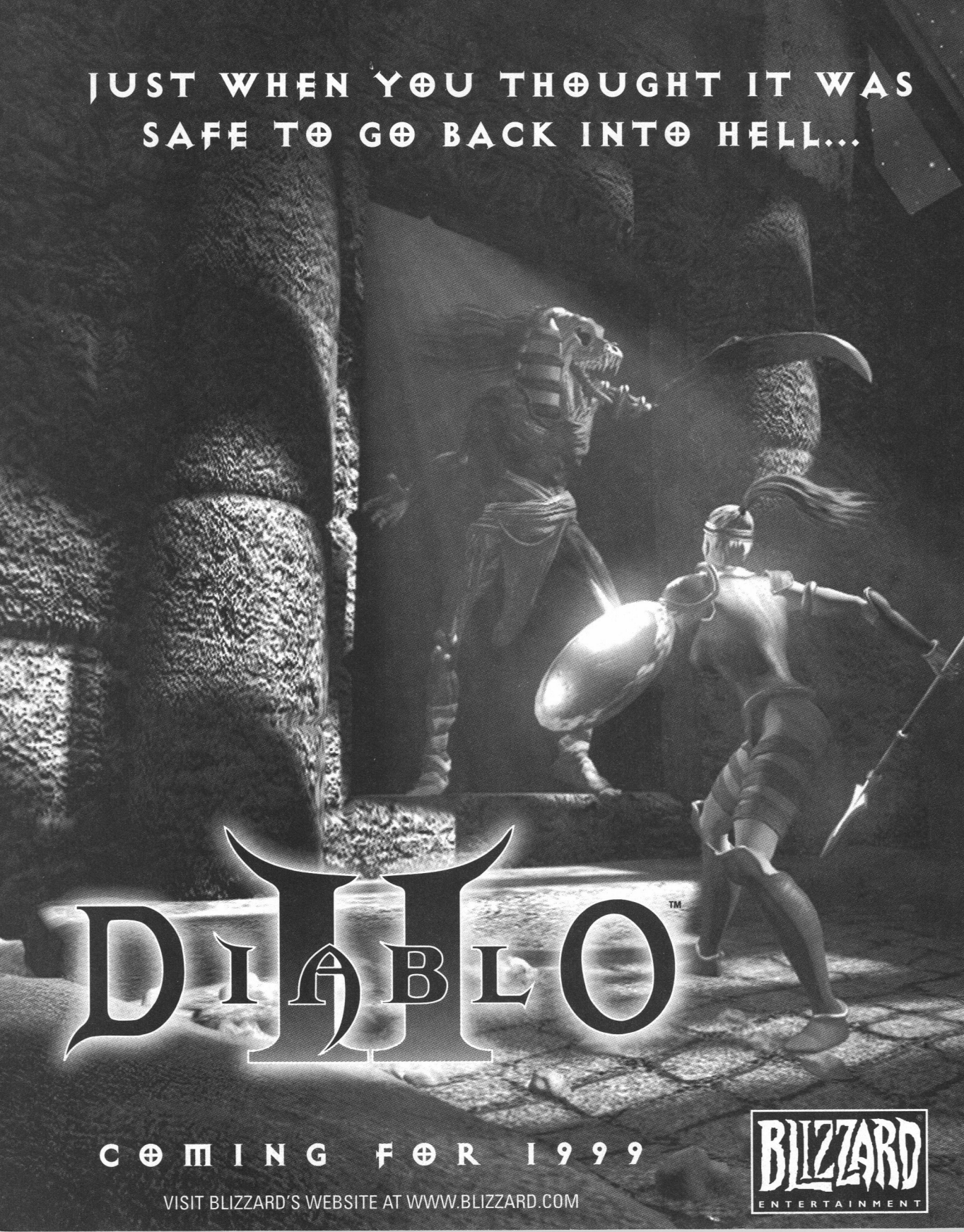
U
unbuildable terrain, 25
Undo command, 17, 28
Unit actions, 104-105
Unit and Hero Settings menu, 31-32, 67-68
Unit Data tables, 143-144
units
changing power settings, 67-68
changing properties of, 17-18
controlling, 61-62, 104-105, 130
computer-controlled, 9, 30, 60, 107
human-controlled, 30, 60
creating, 102
data tables, 143-144
defined, 90
destroying, 104, 123
folders for specific types of, 25-26
moving, 101-102
neutral, 24, 25, 30, 60, 61, 107
removing, 104
selecting, 17
taking over, 123

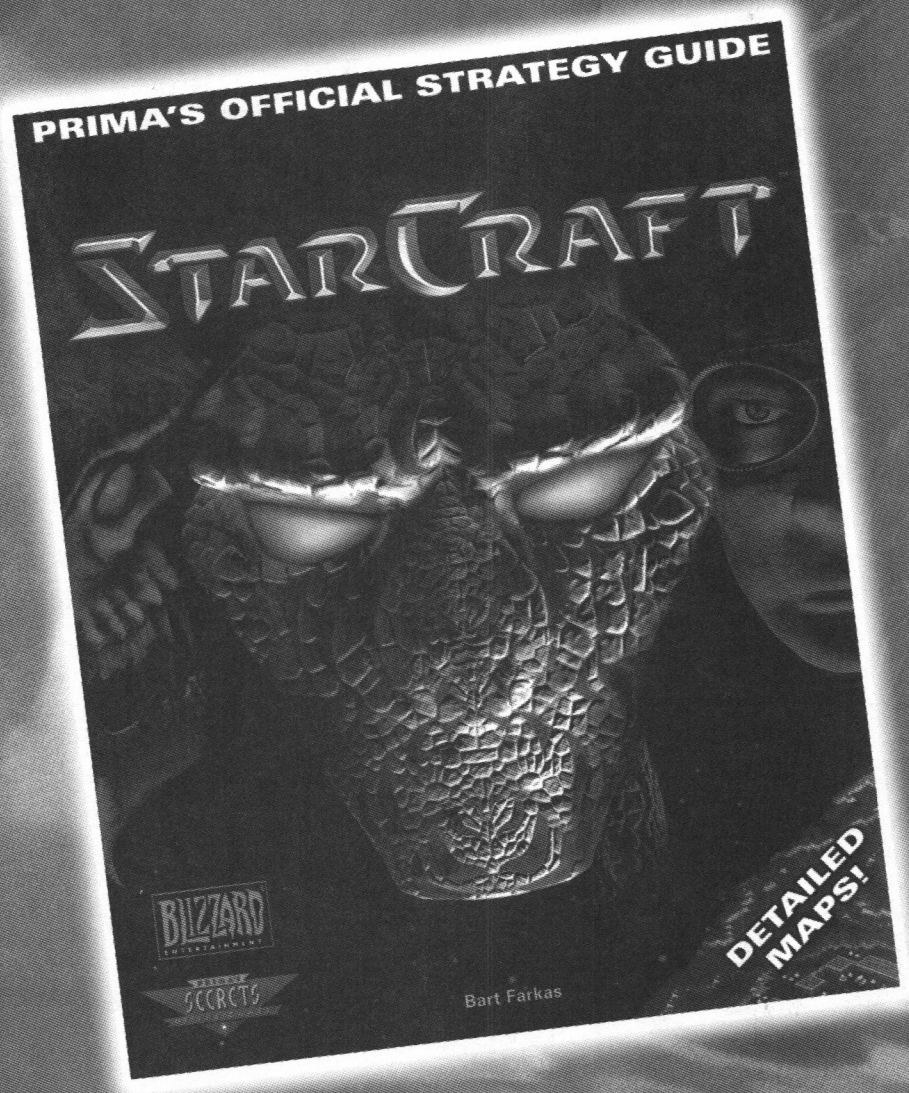
Units menu, 61-62
Unmute Unit Speech action, 103
Unpause action, 107
updates, *Starcraft* and StarEdit, 140
Upgrade Settings menu, 32-33, 68
Upgrades menu, 62-64
Use Map Settings mode, 8-9, 10
User Selectable option, 59

V
"Value This Area Higher" AI, 100
variables, condition, 91
Vespene Geysers, 18, 25, 48, 73
victory
    conditions, 9, 12, 20, 89, 93, 98, 99, 113-118
    triggers, 114-115
View actions, 101-102
volume control, sound file, 103

W
Wait action, 107
Wait commands, 80
Wakka Wakka mission, 13
wall traps, 121
water features, 53
.wav files, 21, 79-80, 102-103
Web site, Blizzard Entertainment, 135, 140
Window menu, 35-36, 142
winning. *See* victory.
worker units, 42

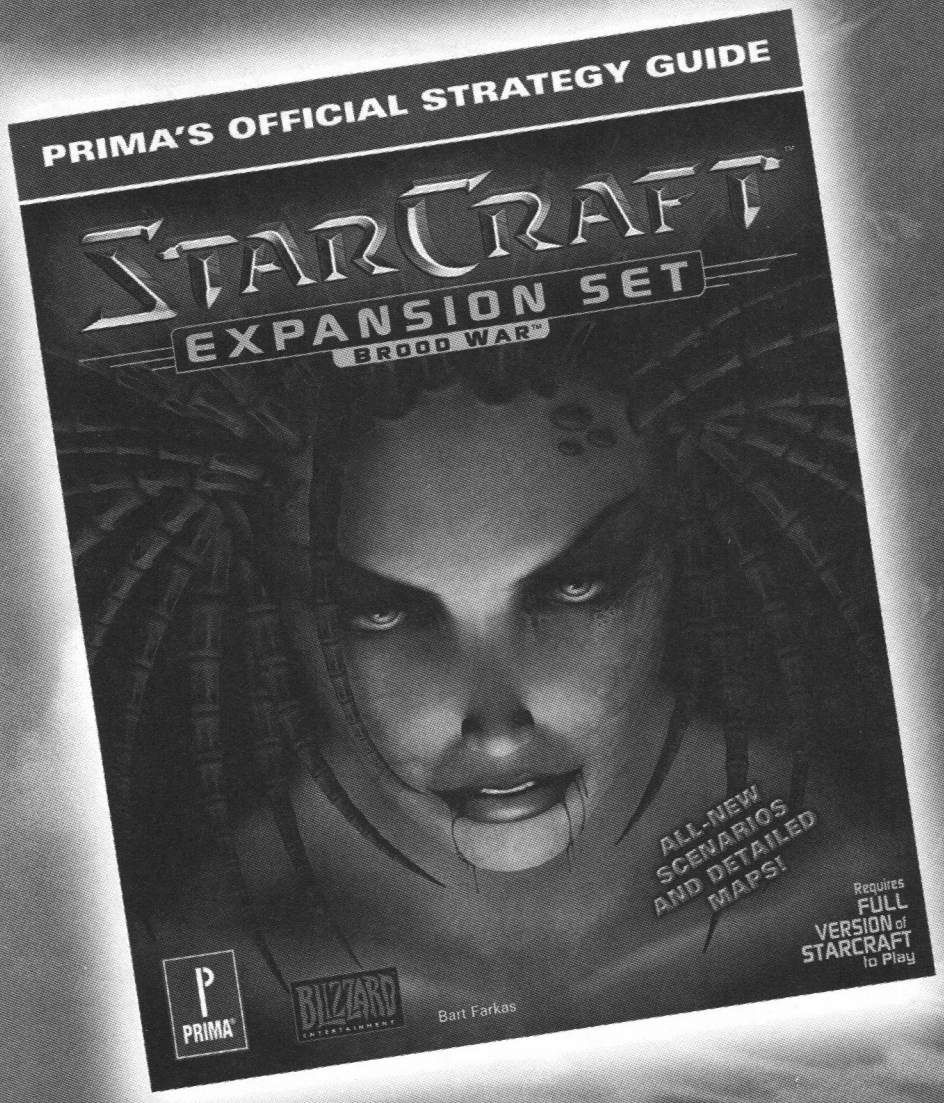
Z
Zerg units and structures, 59, 143, 145


JUST WHEN YOU THOUGHT IT WAS
SAFE TO GO BACK INTO HELL...
DIABLO II™
COMING FOR 1999
VISIT BLIZZARD'S WEBSITE AT WWW.BLIZZARD.COM
BLIZZARD ENTERTAINMENT



to order, call prima at
1-800-531-2343

PRIMA®



coming soon from prima

PRIMA

U.S. $19.99 Can. $27.95 U.K. £15.99

Electronic Entertainment

TURN WAR INTO AN ART FORM WITH STAREDIT!

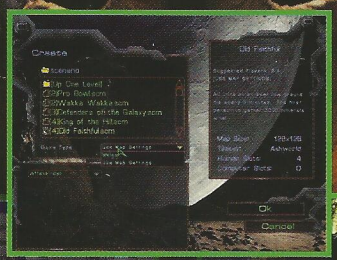
BUILD SCENARIOS AND CAMPAIGNS THAT ARE UNIQUELY YOURS

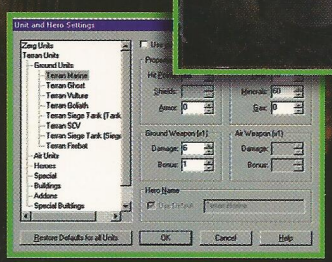
CONSTRUCT BOTH SINGLE- AND MULTIPLAYER MAPS

USE STAREDIT'S TRIGGER SYSTEM TO SPONTANEOUSLY GENERATE NEW UNITS

TRY CAPTURE-THE-FLAG, MAZE, DEATH MATCH, OR EVEN RPG SCENARIOS!

LEARN HOW TO REPLACE GAME SOUNDS WITH ANY SOUND YOU WANT





Complete your Starcraft Library with

Starcraft™: Prima's Official Strategy Guide

Starcraft™ Expansion Set: Brood War: Prima's Official Strategy Guide

Starcraft and Brood War are trademarks and Blizzard Entertainment is a trademark or registered trademark of Davidson & Associates, Inc. in the U.S. and/or other countries.

ISBN 0-7615-1810-X

9 780761 518105 51999

0 86874 51810 0

PRIMA®

The Prima Logo and Prima Publishing ® are Registered Trademarks of Prima Communications, Inc.

www.primagames.com

Platform: PC